NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MULTIRESOLUTION IMAGE RECOGNITION USING THE WAVELET TRANSFORM

by

William M. Peyton, Jr.

June, 1996

Thesis Advisor: Murali Tummala

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
June 1996

3. REPORT TYPE AND DATES COVERED
Engineer's Thesis

4. TITLE AND SUBTITLE
MULTIRESOLUTION IMAGE RECOGNITION USING THE WAVELET TRANSFORM

5. FUNDING NUMBERS

6. AUTHOR(S)
Peyton, William M., Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/ MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/ MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

With the growth of information dissemination over digital communication networks, much research has been devoted to compressing digital image information for efficient transmission. The ability to adjust the desired resolution of an image as the available bandwidth on the network changes allows the user to control the flow of data according to the resources available. In this thesis we integrate multiresolution image compression methods with image recognition techniques to assist in automatic image recognition at several levels of resolution. Features of grayscale and binary images of text characters and aircraft line drawings are described using wavelet transform coefficients, wavelet transform subband energy, and Fourier transform coefficients. Transmission of these features over a digital communication link is simulated, and multiresolution recognition performance in the presence of channel noise is presented.

14. SUBJECT TERMS
Wavelets, Image Compression, Image Recognition

15. NUMBER OF PAGES
90

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

MULTIRESOLUTION IMAGE RECOGNITION USING THE WAVELET TRANSFORM

William M. Peyton, Jr.
Lieutenant, United States Navy
B.S., University of Pennsylvania, 1988

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

and

ELECTRICAL ENGINEER

from the

NAVAL POSTGRADUATE SCHOOL
June 1996

Author: William M. Peyton, Jr.
William M. Peyton, Jr.

Approved By: Murali Tummala
Murali Tummala, Thesis Advisor

Ralph Hippenstiel
Ralph Hippenstiel, Second Reader

Herschel H. Loomis, Jr.
Herschel H. Loomis, Jr., Chairman,
Department of Electrical and Computer Engineering

TABLE OF CONTENTS

ABSTRACT

With the growth of information dissemination over digital communication networks, much research has been devoted to compressing digital image information for efficient transmission. The ability to adjust the desired resolution of an image as the available bandwidth on the network changes allows the user to control the flow of data according to the resources available. In this thesis we integrate multiresolution image compression methods with image recognition techniques to assist in automatic image recognition at several levels of resolution. Features of grayscale and binary images of text characters and aircraft line drawings are described using wavelet transform coefficients, wavelet transform subband energy, and Fourier transform coefficients. Transmission of these features over a digital communication link is simulated, and multiresolution recognition performance in the presence of channel noise is presented.

ACKNOWLEDGEMENTS

Completion of this thesis is the end of a long process begun in June of 1993 culminating in the awarding of my degrees. Throughout this process, I have received support, encouragement, and occasional motivational nudges from my wife, Vanessa, my mother, Grace, and my sisters, Kathryn and Leslie. I would like to thank all of them for helping me to achieve a goal which seemed so far away, and at times, unreachable. Thank you.

I would also like to acknowledge the creators of the Wavelab .700 Matlab programs at Stanford University. David Donoho, Iain Johnstone, and others have created an excellent set of tools which allows new students of signal and image processing with wavelets to develop new skills and understanding far more rapidly than possible otherwise. This toolbox, along with others created at Rice University and at the University of South Carolina, was immensely helpful in my research.

I. INTRODUCTION

The growth of information exchange over digital communication networks has made image compression one of the most important research topics in recent years. The ability to adjust the desired resolution of an image as the available bandwidth on the network changes allows the user to control the flow of data according to the resources available. In this thesis a scheme for integrating multiresolution image compression with automatic image recognition techniques is presented. It is shown that this method can achieve fast and accurate image recognition at varying resolution levels.

Figure 1 shows a model of a digital communications link which shall be used in the rest of the thesis. The available bandwidth on the channel varies with the traffic on the

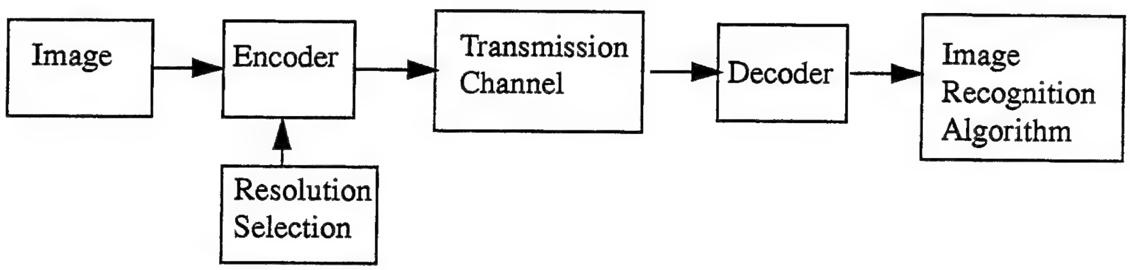


FIGURE 1. Multiresolution Transmission Scheme, After Ref 9, p. 33

channel. The user selects one of five resolution levels for the image, adjusting the desired resolution as traffic varies. Normally, the user selects the lowest resolution level to minimize the load on the channel. One of the important features in this scheme is the ability to recognize images at a variety of resolution levels. This allows the user to use low resolution images to identify images of particular interest and then select higher levels of resolution only for these images, thus maximizing the utility of the available bandwidth. A typical scenario in which this model is applicable is transmission from a remote sensor where the processing power and memory storage available onboard is insufficient to per-

form image recognition. Another example is searching an image archive at a remote site.

In this thesis, multiresolution image compression with the wavelet transform is integrated with image recognition algorithms to perform multiresolution image recognition of text characters. The processing overhead due to image registration required to perform recognition on images which have undergone linear translation can be reduced by using Fourier transform coefficients as elements of the feature vectors. The memory and the bandwidth required to perform recognition are reduced by using the energy in each subband of the wavelet transform as elements of the feature vectors.

Chapter II introduces the fundamentals of image recognition and shows the importance of the feature selection problem in designing a recognition system. Chapter III provides background for multiresolution signal decomposition with the wavelet transform. Chapter IV presents results of the proposed multiresolution image recognition algorithm using text characters. Chapter V summarizes the results and suggests areas of future research. Computer code used in this thesis is presented in the Appendix.

II. IMAGE RECOGNITION

This chapter discusses the digital image representation and recognition scheme which is the backbone of the work presented in this thesis. The scheme can be described in terms of a system of image processing functions which must be performed to convert an image of a real-world object into a form which a computer can analyze and classify as shown in Figure 2. We shall use scanned images of text characters to test our recognition algorithm, but the discussion in this chapter holds for any black and white image.

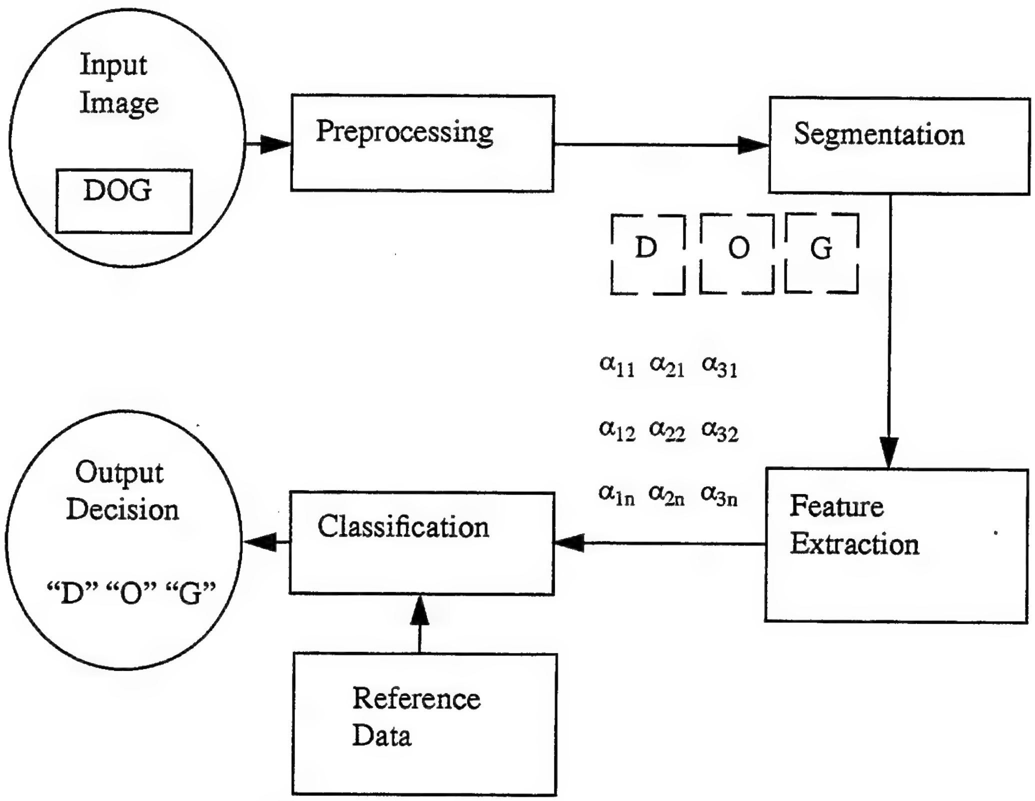


FIGURE 2. Image Recognition System Diagram After Ref 1, p. 8

The scheme begins with an image sensor which captures the image. If the sensor is an analog device, such as a film camera, the image must be digitized and converted to a grayscale image, often by using a scanner. For each small area of the image, known as a picture element or pixel, the scanner converts the analog image to a number which represents the relative intensity of light in that pixel. A typical scanner may divide the

intensity spectrum between black (zero brightness) and white (maximum brightness) into 256 or more distinct shades of gray. This is known as a grayscale image.

The grayscale image is then passed to a segmentation routine, which separates objects in the image by applying a threshold to the image, using the fact that distinguishable objects' pixel values differ markedly from the pixel values of the surrounding environment. In Figure 2, the objects are the individual letters in the word "DOG." The objects are then input to a feature extraction routine, which distills important information from the image, called features. The individual features α_1, α_2, etc., are combined together into an n-dimensional feature vector $[\alpha_1, \alpha_2, \ldots \alpha_n]$, which is then passed to a classification routine. The choice of features determines the length of the feature vector. In Chapter IV, we shall use feature vectors which vary from 9 to over 16,000 elements. The classification routine compares the feature vector obtained from the image currently being classified, known as the test image, with the feature vectors obtained previously from reference images. The classifier associates the test image with one of the reference images, and outputs the decision thus made.

The performance of an image recognition system is expressed using the percentage of correctly classified images. For a character recognition system, this is the rate at which the classifier outputs a "D" when a "D" was input, for example; if the classifier declares that the image was anything other than a "D," an error has been made. There are many possible sources of errors. To illustrate this, a detailed discussion of each image processing function is required.

A. IMAGE ACQUISITION

The two elements necessary to create an image of a real-world object in a form readable by a computer are a physical device that is sensitive to electromagnetic energy, such as a camera or charge-coupled device, and a digitizer, which converts the information into a discrete set of numbers [Ref 1, p. 10]. Consider a flatbed image scanner, for example, which was used to create the reference images in this thesis. A scanner sensor, like many photocopier sensors, consists of a line of silicon imaging elements, called photosites, which

produce an electrical voltage output proportional to the intensity of the light reflected from the original image as it is scanned [Ref 1, p. 12]. The sensor outputs a set of voltages corresponding to the distribution of brightness in the original image.

The set of individual voltages is formed into an NxM array, each element of which is a pixel. This array is then fed to a digitizer, which converts the voltage levels to a discrete set of numbers corresponding to the voltage in a particular area of the image. These numbers, called the grayscale values, are limited to a particular range [0, J]. Generally, J is a power of 2. These three parameters, N, M, and J, determine the two factors which govern the resolution (degree of discernible detail) of the picture obtained from the digitization process: spatial resolution and gray level quantization.

Note that there is a tradeoff between resolution and the memory required to store the digital image: the finer the resolution, the greater the storage requirement. For example, an image comprised of 256 x 256 pixels (65,536 pixels total) quantized into 128 gray levels (requiring at least 7 bits per pixel) takes up approximately 64 kilobytes of memory. The same image using 1024 x 1024 pixels and 256 gray levels requires over 1 megabyte of storage [Ref 1, p. 33]. In addition to the added burden on memory resources, more pixels means longer processing times at each stage of the image recognition system, at least until feature extraction. Determining the proper tradeoff between resolution and storage requirements is one of the fundamental design decisions facing the image processing system designer. For the purpose of image recognition, the desired resolution is the minimum required to extract feature vectors sufficiently distinct to provide accurate classification. If the original image does not have sufficient resolution, there are numerous ways to enhance the image by additional processing.

B. PREPROCESSING

These methods, which fall into the pre-processing block in the system diagram, include filtering, edge enhancement, and contrast enhancement. Each of these is designed to amplify crucial details present in the image while suppressing noise and undesired image elements.

Filtering can be done in either the spatial or the frequency domain. In the spatial domain, filtering amounts to convolving the image with a window chosen for its spectral properties. Examples are shown in Figure 3.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{9} \times \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Low Pass Filter — High Pass Filter — Sobel Horizontal Gradient Filter

FIGURE 3. Examples of Spatial Filters. After Ref 1, p. 195-200

The low pass filter in Figure 3 computes the average of pixels in a 3x3 neighborhood around a pixel in the center. This suppresses noise, but blurs sharp edges. The high-pass filter enhances edges, but magnifies noise. The Sobel filter computes the gradient in the y direction at the center pixel. When this quantity is added to (subtracted from) the original pixel value, contrast between the value of the center pixel is increased (decreased). If the pixels in this 3x3 neighborhood are uniform, this operation has no effect. However, if the neighborhood contains a horizontal edge, the value of the center pixel will be changed considerably, thus increasing the contrast between this pixel and background pixels. The Sobel vertical filter is the transpose of the horizontal filter.

Filtering in the frequency domain is performed by computing a two-dimensional transformation of the image, such as the Fourier transform, and then assigning various weights to the components in the transform domain to accomplish a desirable enhancement. Just as in the spatial domain, there is a tradeoff between reducing noise and reducing the contrast in the vicinity of edges. Weighting low-pass frequencies relatively more than high pass frequencies reduces noise, but blurs the sharp edges in the image. Conversely, weighting higher frequencies more than lower frequencies enhances edges at the cost of magnifying noise. Finding the optimal mix of enhancements for a given image processing application must be performed on a case by case basis.

C. SEGMENTATION

After the image has been enhanced, the next step is to distinguish between objects and surrounding background and to differentiate among objects. The most basic segmentation method is grayscale thresholding. The segmentation algorithm converts the grayscale image to a binary image, where every pixel below the threshold is assigned a value of 0, and every pixel above the threshold is assigned a 1. If one has no prior knowledge of the type of image to be processed, one technique of selecting a threshold is histogram partitioning, as shown in Figure 4.

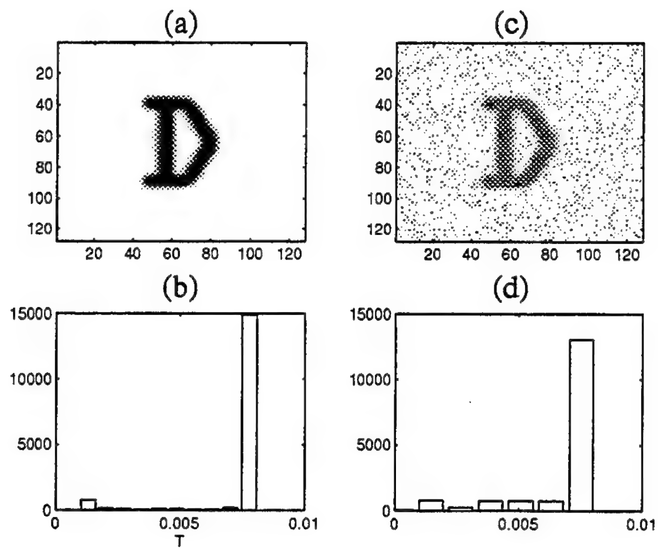


FIGURE 4. Thresholding Using Image Histogram

In this example, we see an image of a text character in Figure 4a. A histogram of its pixel values is shown in Figure 4b. There is a clear break between one cluster of pixel values and the other, making the choice of the threshold at T obvious. Unfortunately, for many images, such as the text character shown in Figure 4c, the choice is often not as clear due to noise in the scanning process or the nature of the object and background. This character's histogram is shown in Figure 4d. Note that there is no clear threshold for this poorly processed image. Nevertheless, for the purpose of character recognition, the choice of the threshold for a particular image is not as important as consistency between

thresholding the reference image and the test images. So long as both the reference image and the test image use a similar threshold, the feature vectors extracted from each should be similar.

D. FEATURE EXTRACTION AND IMAGE REGISTRATION

After segmentation, the binary image is passed to a feature extraction routine. Image features are quantities which carry information about the object in the image, such as size, texture, color, etc. They may also contain transform domain information such as the energy in a particular band of frequencies. Images may be formed into classes based on the similarity of their feature vectors. The choice of features which efficiently and accurately describe the various classes of images is known as the feature selection problem. In general, spatial domain features are sensitive to variations in image translation, rotation, and scale changes [Ref 1, p. 501]. If one uses spatial domain features, one must compensate for spatial variances by registering the image.

Registration is the process by which one corrects for relative translational shifts, rotational shifts, and resolution differences from one image to another [Ref 2, p. 562]. Image translation can be compensated for by calculating the correlation between the reference image $X_r(m,n)$ and a test image $X_t(m,n)$ for all possible shifts of the object within the image. In its simplest form, this measure is defined as

$$R(m, n) = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} X_r(m, n)X_t(m-j, n-k)}{\left[\sum_{m=1}^{M}\sum_{n=1}^{N} X_r^2(m, n)\right]^{\frac{1}{2}}\left[\sum_{m=1}^{M}\sum_{n=1}^{N} X_t^2(m-j, n-k)\right]^{\frac{1}{2}}}, \quad \text{(EQ 1)}$$

where (m,n) are pixel positions in an MxN image. There are several problems with this method, however. First, the correlation may be relatively broad, i.e., having no single sharp peak in the correlation function. Second, noise may mask the true peak. Finally, registration is computationally expensive, especially if the relative motion between images is significant [Ref 2, p. 566].

Another solution to the registration problem is to find an alternate representation of the image which is invariant to translation, rotation, contraction, or reduction. One example of this is to take the Fourier transform of the image and use the Fourier coefficients to create feature vectors. The magnitude of Fourier transform coefficients in a rectangular coordinate system are invariant to linear translation, i.e., the coefficients change due to translation of the object in the spatial domain by changing in phase, but not in magnitude [Ref 1, p. 95]. We will use this property to our advantage in Chapter 4 by taking the Fourier transform of wavelet transform coefficients to obtain multiresolution translation invariant recognition. However, Fourier coefficients in a rectangular coordinate system are not invariant to translations in rotation [Ref 1, p. 99]. Rotating an image by an angle φ rotates the Fourier transform by the same angle. If one first converts the image to polar coordinates, one can obtain a polar Fourier transform, for which the magnitude of the coefficients is invariant to rotation. Unfortunately, one loses linear translation invariance in the process. Since the focus of this thesis is on the recognition of text characters, which are generally not subject to rotational changes, we shall not address rotational variance further.

Once the feature vector has been obtained, it is compared with feature vectors from various reference images which represent the different classes of the image. The feature vectors of the reference images occupy an n-dimensional Euclidean space, called the feature space, as shown in Figure 5. Each reference feature vector $\mathbf{r}_i$ represents a point in the feature space. One logical and simple way to classify test images is to compute the Euclidean distance between each of the reference feature vectors $\mathbf{r}_i$ and the test feature vector $\mathbf{t}$:

$$dist(r_i,t) = \sqrt{(r_i - t)^T (r_i - t)} \quad \text{(EQ 2)}$$

The test image is then associated with the reference image which has the smallest distance measure. If the net effect of all the various forms of image noise can be modeled as additive white Gaussian noise, it can be shown that the minimum distance classifier is optimum [Ref 1, p. 581].

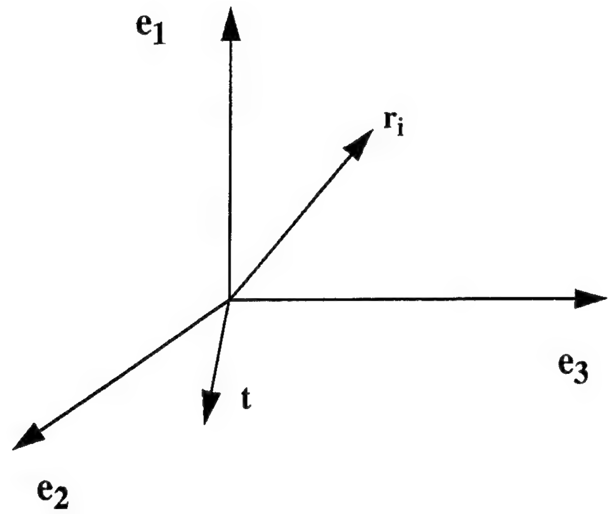


FIGURE 5. Feature Vectors And Euclidian Distance in Feature Space

Another possible classification method is a neural network based approach, which has great utility when the statistical properties of the pattern classes are unknown. A multilayer network is trained with the reference images using a learning technique such as the backpropagation algorithm [Ref 1, pp. 595-602]. If the network has learned the proper classification for each reference image, it should be able to correctly classify a test image. Unfortunately, analyzing the performance due solely to the influence of the selection of image features is difficult. In this thesis, we shall use the minimum distance classifier.

We have seen that the performance of an image recognition system depends on a chain of processes, each one of which is essential to the proper functioning of the system. Feature selection is an important part of the recognition system. The wavelet transform, which supports a multiresolution approach to the feature selection problem, is presented in the next chapter.

III. MULTIRESOLUTION IMAGE COMPRESSION AND THE WAVELET TRANSFORM

Multiresolution signal decomposition can be used as a form of flow control in a packet switched network by allowing a user to adjust the desired resolution of an image based on the available bandwidth in a communications channel [Ref 9, p. 1]. We will show how the wavelet transform allows us to perform multiresolution image compression and show how this can be integrated with the algorithms from Chapter II to recognize images at different levels of resolution.

A. MULTIRESOLUTION IMAGE COMPRESSION AND RECOGNITION

Figure 6 shows the model of a multiresolution image compression and recognition scheme over a digital transmission channel (packet switched network) used in this thesis.

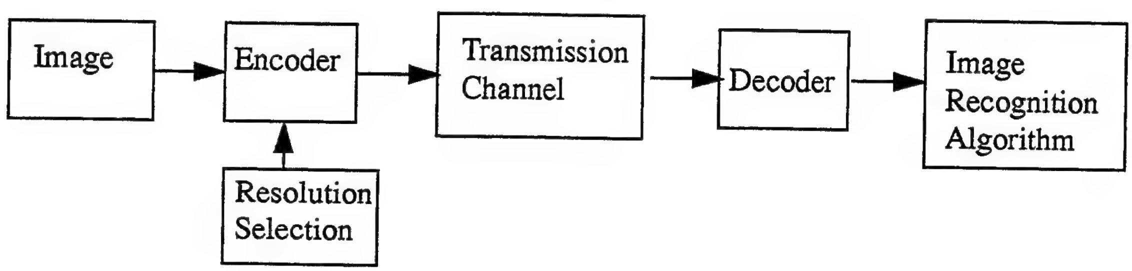
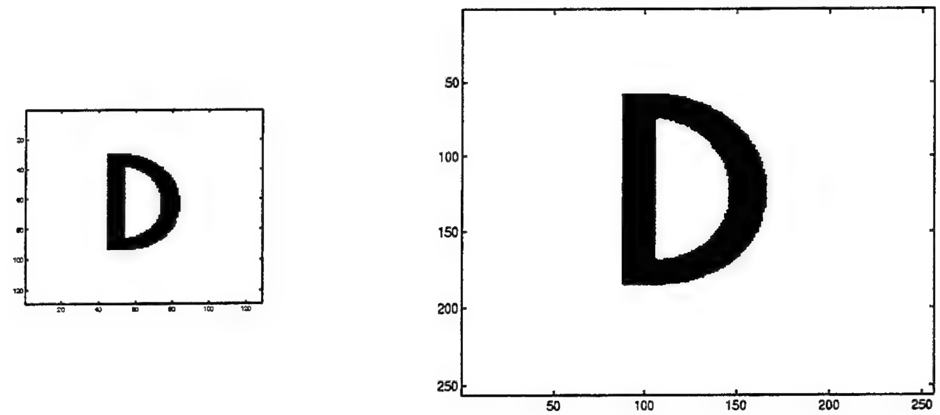


FIGURE 6. Multiresolution Transmission Scheme, After Ref 9, p. 33

In this scheme, the user desires to transmit images of unknown content from a remote site as quickly as possible. This is often a problem when searching an image archive. The available bandwidth in a packet switched network varies with the requested load on the channel. Multiresolution image coding allows the user to select the desired amount of detail in an image to be transmitted depending on the current load as measured by the packet delay through the channel. If the channel is congested, the user will want to transmit images at the lowest level of resolution from which they can be recognized. If the user decides that the image is of particular interest, it can be enhanced by adding additional detail with further transmissions [Ref 9, p. 26].

To determine rapidly and accurately which images are of interest, the user requires a way to compare images at different levels of resolution as shown in Figure 7. As stated in Chapter II, registering images at different levels of resolution is computationally expensive, so we require a way to compare images without first registering them.



7a 7b

FIGURE 7. Two Images of an Object at Different Scales

The solution is to use the orthogonal projection of the image in Figure 7b onto the space spanned by the image in Figure 7a. We obtain the orthogonal projection simply by taking the wavelet transform of the image in Figure 7b and discarding the smallest scale wavelet transform coefficients, then performing the inverse wavelet transform [Ref 11, p. 314-320]. Alternatively, we can perform image recognition in the wavelet domain, using wavelet coefficients as image features in our recognition scheme. Before showing how this can be done, we first present a brief overview of the wavelet transform.

B. ORTHOGONAL BASIS FUNCTION EXPANSIONS

We desire to decompose a signal x(t) using elementary functions $\phi_1(t)$, $\phi_2(t)$,... so that we may discern information present in the signal which may not have been obvious in its original form. If every function contained in a vector space V can be written as a linear combinations of linearly independent vectors $\phi_k(t)$ which span V, i.e.

$$x(t) = \sum_k c_k \phi_k(t), \qquad \text{(EQ 3)}$$

then the set $\phi_1(t)$, $\phi_2(t)$,... forms a basis for V [Ref 5, p. 78]. The c_k's are the transform coefficients, i.e., they are the projection of the function x(t) onto the basis function $\phi_k(t)$. If the basis functions are orthogonal, we can compute this projection by taking the inner product of the signal x(t) with the basis function $\phi_k(t)$ [Ref 10, p. 245]. We define the inner product operation for continuous signals as

$$c_k = \langle x(t), \phi_k(t) \rangle = \int_t x(t)\phi_k(t)dt \qquad \text{(EQ 4)}$$

and for a discrete signal of length N,

$$c_k = \sum_{n=1}^{N} x(n)\phi_k(n). \qquad \text{(EQ 5)}$$

The set $\phi_1(n)$, $\phi_2(n)$,... is said to be orthonormal if

$$\langle \phi_i(n), \phi(n)_j \rangle = \delta_{ij} \qquad \text{(EQ 6)}$$

where δ_{ij} is the Kronecker delta function.

One well-known form of signal decomposition which has these properties is the discrete Fourier transform, in which the $\phi_k(n)$ are complex exponentials at a single frequency. Each transform coefficient, therefore, is simply the signal content at the frequency of the exponential. Any discrete signal can be represented as a weighted sum of these basis functions, i.e.,

$$x(n) = \frac{1}{N}\sum_{k=1}^{N} c_k e^{\frac{j2\pi kn}{N}} \qquad \text{(EQ 7)}$$

where each c_k is the coefficient at the k-th frequency,

$$c_k = \sum_{k=1}^{N} x(n) e^{-\frac{j2\pi kn}{N}}. \qquad \text{(EQ 8)}$$

This is equivalent to a filter bank formation as shown in Figure 8a [Ref 8, p. 800].

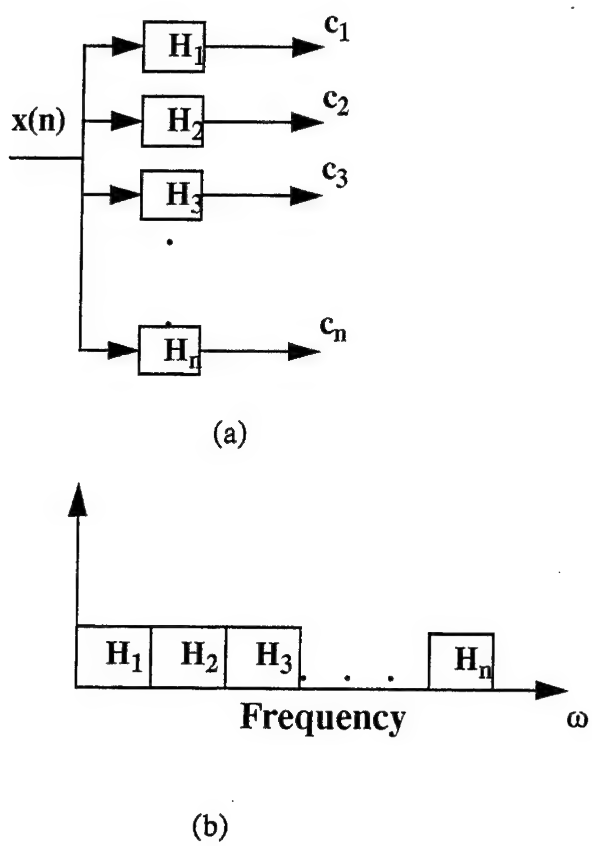


FIGURE 8. Fourier Transform as Ideal Orthogonal Filter Bank Tiling the Frequency Axis

Each basis function in the Fourier transform corresponds to a non-overlapping bandpass filter. The output of each of these filters is the content of the signal x(n) in each frequency band. The division of the frequency axis into contiguous nonoverlapping sections is known as tiling, as shown in Figure 8b.

Frequency is not the only important characteristic of a signal. An equally important feature, particularly for images, is scale. Scale is the amount of time or space over which a particular signal component is significant. Note that while the Fourier coefficients locate signal energy well in frequency, the time or space resolution is poor. Wavelet analysis does not suffer this limitation and provides a way to localize signals in both scale and space.

C. 1-D WAVELET TRANSFORM

We start with the multiresolution formulation of Mallat for one-dimensional signals [Ref 6, p. 6]. Any signal can be decomposed into low-pass and high-pass signals. Formally, a vector space V_{j+1} is comprised of two spaces: V_j, which contains all low-pass functions, and W_j, which contains all high-pass functions, or

$$V_j \oplus W_j = V_{j+1}. \quad \text{(EQ 9)}$$

The spaces V_j and W_j are called orthogonal complements of each other because they are nonoverlapping, and they combine to span V_{j+1}. One can decompose a signal x(n) which is comprised of frequencies in the range $[0, \pi]$ by one set of basis functions which spans the range $V_j = [0, \pi/2]$ and another set which spans the range $W_j = [\pi/2, \pi]$, as shown in Figure 9a. If we extend this idea by successively dividing the range $[0,\pi/2]$ into smaller and smaller subspaces, as shown in Figure 9b, we see that any subspace V_j contains an infinite number of subspaces V_{j-1}, V_{j-2}, etc. Each of these spaces also has a complementary space W_{j-1}, W_{j-2}. The space which contains all square summable functions, known as L^2, contains all subspaces V_j. This relation is summarized below:

$$V_j \subseteq V_{j+1} \subseteq V_{j+2} \subseteq V_\infty = L^2. \quad \text{(EQ 10)}$$

Mallat showed that the spaces V_j, V_{j-1}, etc. are spanned by dilations and integer translations of a single scaling function $\phi_j(n)$ (a low-pass function), and the spaces W_j are similarly spanned by dilations and integer translations of $\psi_j(n)$ (a high- pass function) [Ref 6, p. 6]. Because V_{j-1} and W_{j-1} are subspaces contained wholly within V_j, $\phi_j(n)$ and $\psi_j(n)$ at scale j can be expressed in terms of a filter and the functions themselves at scale j-1. If h(n) is a low-pass filter and g(n) is a high-pass filter, then

$$\phi_j(n) = \sum_i h(i)\phi_{j-1}(2n-i)\text{, and} \quad \text{(EQ 11)}$$

$$\psi_j(n) = \sum_i g(i)\phi_{j-1}(2n-i). \quad \text{(EQ 12)}$$

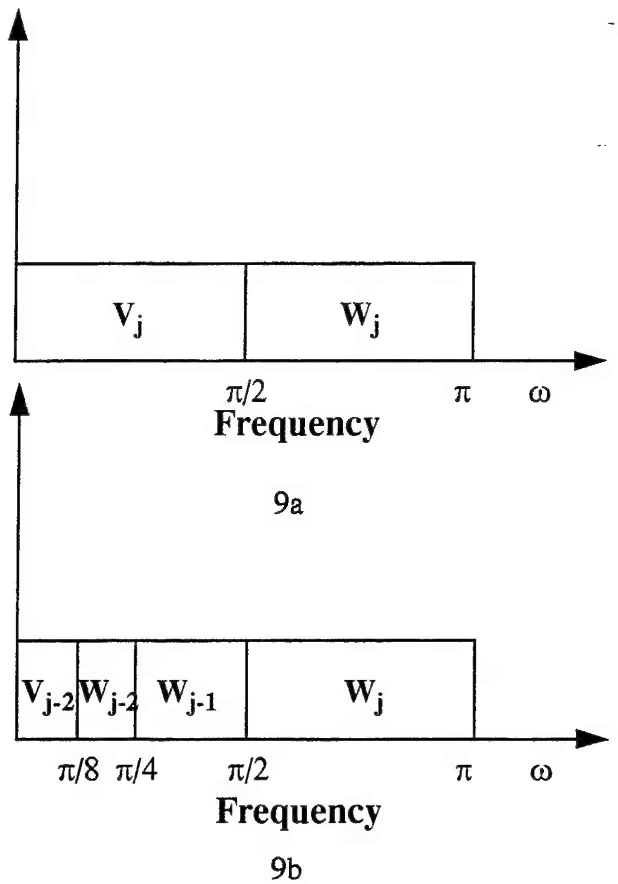


FIGURE 9. Ideal Multiresolution Signal Decomposition

where $\phi_{j-1}(2n\text{-}i)$ is the ith translation of $\phi_{j-1}(2n)$, which is a basis function for scale j-1, and $\phi_j(n)$ is a basis function for scale j.

If we make the translations and dilations of the scaling functions orthogonal, then it can be shown that any function x(n) can be decomposed into high-pass and low-pass signal components, i.e.,

$$x(n) = \sum_k c_k \phi_k(n) + \sum_k d_k \psi_k(n), \tag{EQ 13}$$

where

$$c_k = \langle x(n), \phi_k(n) \rangle \tag{EQ 14}$$

and

$$d_k = \langle x(n), \psi_k(n) \rangle. \qquad \text{(EQ 15)}$$

Here c_k is the inner product of x(n) and $\phi_k(n)$, and d_k is the inner product of x(n) and $\psi_k(n)$, and k indicates the translation index [Ref 6, p. 9]. Further, it can be shown that in order for the reconstruction of the signal to be perfect, h(n) and g(n) must form a quadrature mirror filter (QMF) pair [Ref 6, p. 15]. These fundamental results of wavelet analysis were developed independently in the field of subband coding.

D. IMPLEMENTATION WITH QUADRATURE MIRROR FILTERS

The basic building block of discrete time wavelet analysis is the quadrature mirror filter pair. This set of filters allows us to decompose a signal into low-pass and high-pass signal components, decimate the resulting signals, and then to reconstruct the original signal from these components perfectly. A block diagram of this scheme is shown in Figure 10, and the frequency response of a quadrature mirror filter pair is shown in Figure 11.

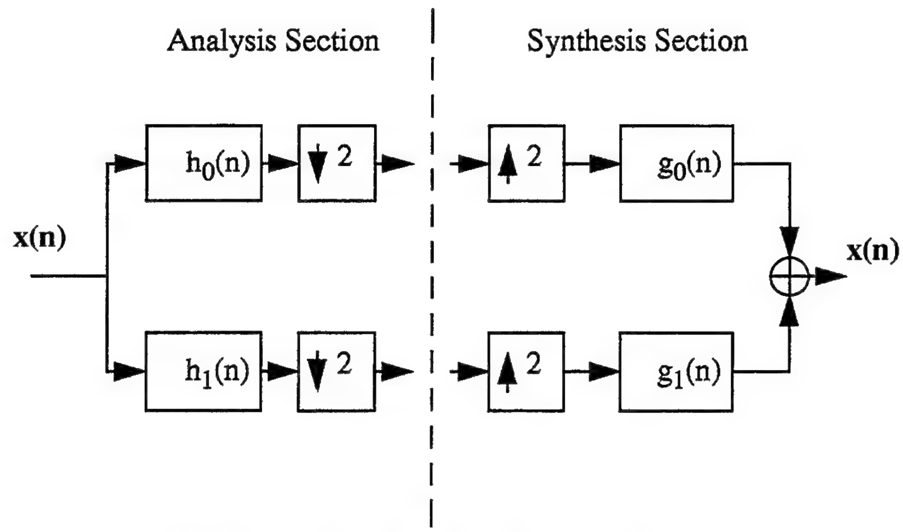


FIGURE 10. Block Diagram of Analysis And Synthesis Sections of a Quadrature Mirror Filter Bank

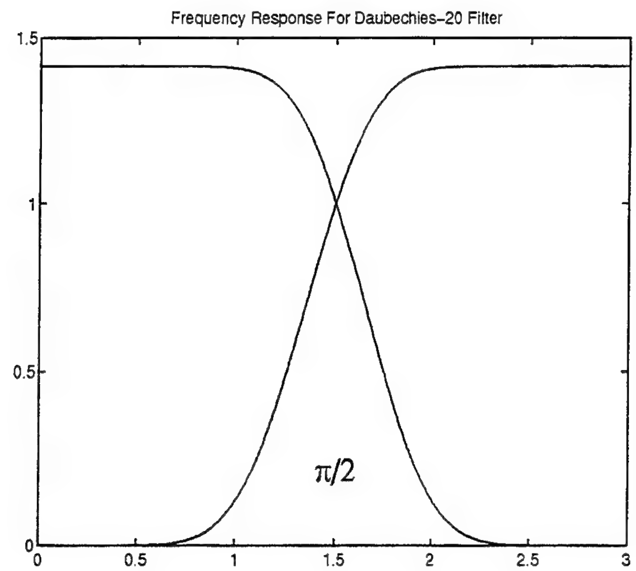


FIGURE 11. Frequency Response Of QMF Pair

If we continue to apply signal decomposition with quadrature mirror filters on the low-pass signal component, we obtain Mallat's scheme for the discrete wavelet transform [Ref 6, p. 11]. The input signal x(n) is successively low-pass and high-pass filtered followed by downsampling. The resulting functions c_j and d_j represent the low-pass "coarse" and high-pass "detail" signals, respectively, at each scale. From the previous discussion, c_j is a function in space V_j, and d_j is a function in space W_j. Figure 12 shows a representation of this scheme, which is known as the fast wavelet transform.

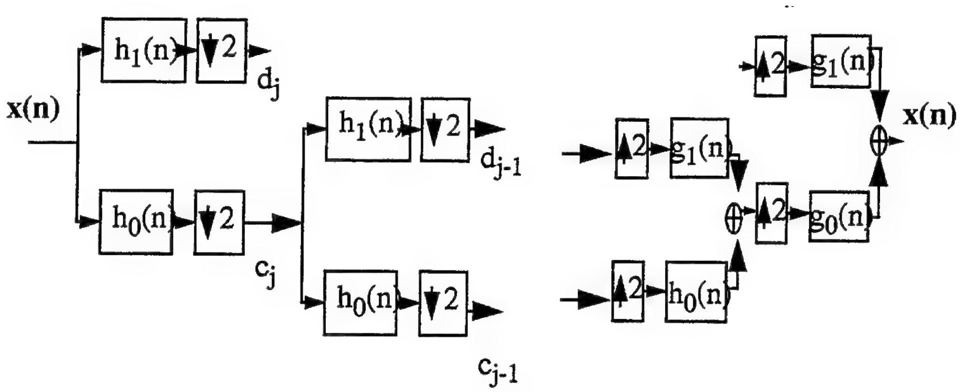


FIGURE 12. Fast Wavelet Transform Block Diagram

E. TWO-DIMENSIONAL FAST WAVELET TRANSFORM

The one-dimensional discrete-time wavelet transform can be extended to two dimensions by assuming that the scaling and wavelet functions are separable, i.e.,

$$\phi_j(m, n) = \phi_j(m)\phi_j(n). \qquad \text{(EQ 16)}$$

At each scale j, we need three wavelet functions, corresponding to the cases where we have low-pass frequencies in m and high-pass frequencies in n, high-pass in m and low-pass in n, and high-pass in both variables, respectively. These are written as

$$\psi_j 1(m, n) = \phi(m)\psi(n) \qquad \text{(EQ 17)}$$

$$\psi_j 2(m, n) = \psi(m)\phi(n) \qquad \text{(EQ 18)}$$

$$\psi_j 3(m, n) = \psi(m)\psi(n) \qquad \text{(EQ 19)}$$

If $M = 2^J$, which we shall assume throughout this thesis, an M x M image will have J orthogonal scales. As shown in Figures 13 and 14, the two-dimensional wavelet transform decomposes the signal into four subbands, k= 1, 2, 3, and 4, at each scale j. The k = 1 subband, which is low-pass in both m and n directions, is then decomposed further at one scale lower. This process may be repeated until the signal at the largest scale is a single coefficient. Dividing an image into components of differing scale is similar to tiling the frequency axis as discussed in section III.B. Subband energy is a measure of the signal

content in a particular tile. The energy in a given K x K subband E_{jk} is the squared sum of the wavelet coefficients in that subband

$$E_{jk} = \sum_{i}^{K} \sum_{l}^{K} c^2_{iljk},\ j \in 1, 2 \ldots J; k \in 1, 2, 3, 4. \qquad \text{(EQ 20)}$$

In Chapter IV, we will show that subband energy values can be used as features in an image recognition scheme.

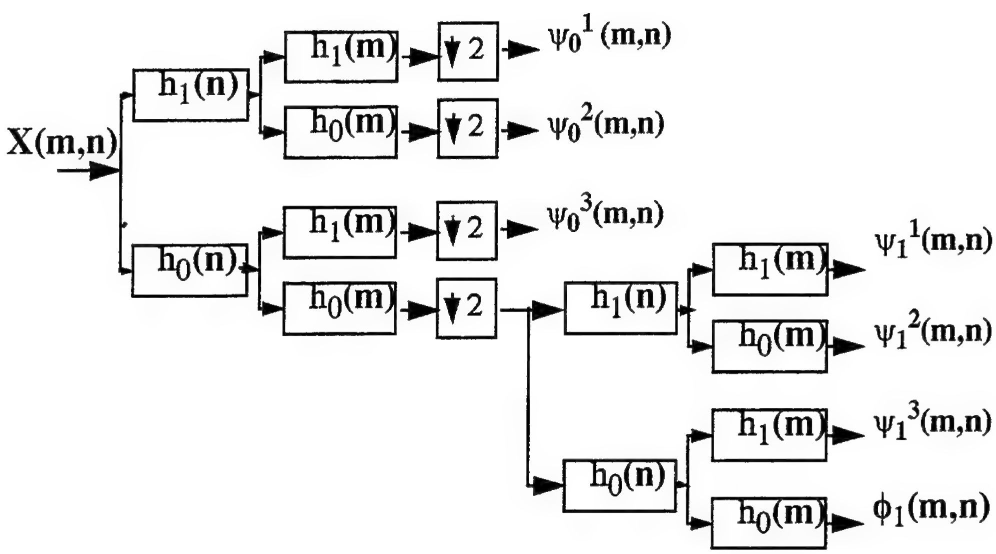


FIGURE 13. Two-Dimensional Fast Wavelet Transform

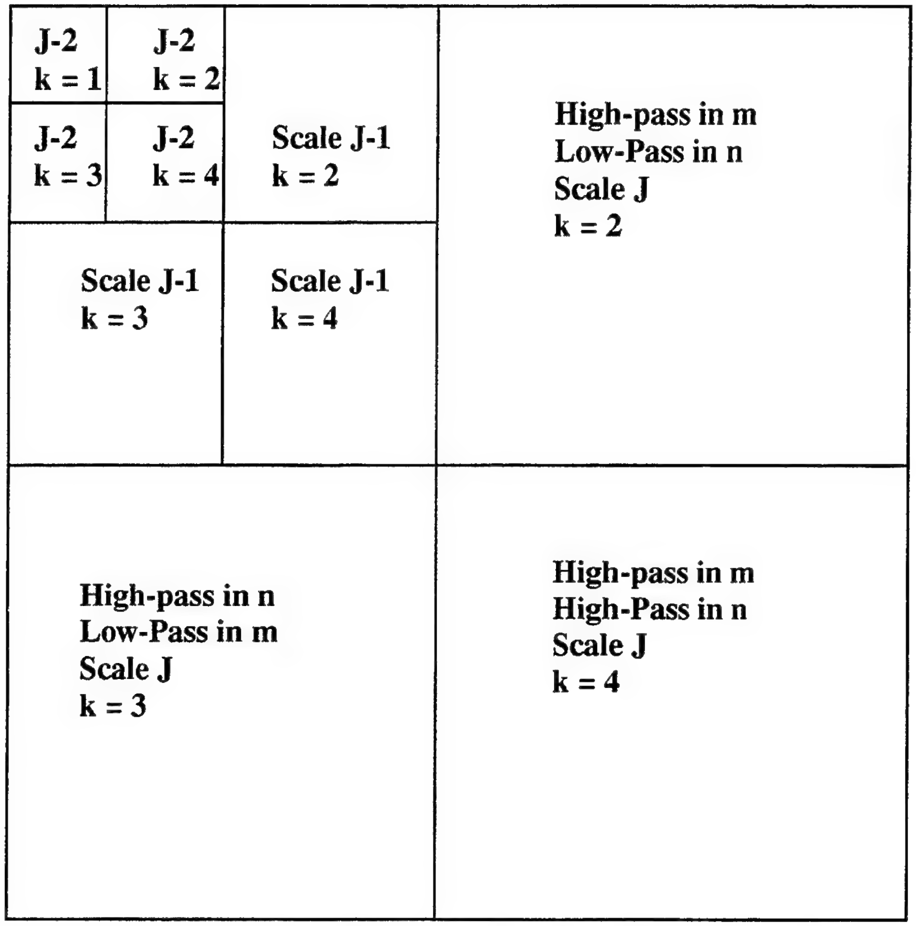


FIGURE 14. Decomposition of Image Into Components In Orthogonal Subspaces

F. MULTIRESOLUTION CODING USING THE WAVELET TRANSFORM

The two dimensional wavelet transform can be used to create a multiresolution image compression scheme [Ref 7, p. 26]. Figure 15 shows the tiling used for the proposed multiresolution image compression recognition scheme.

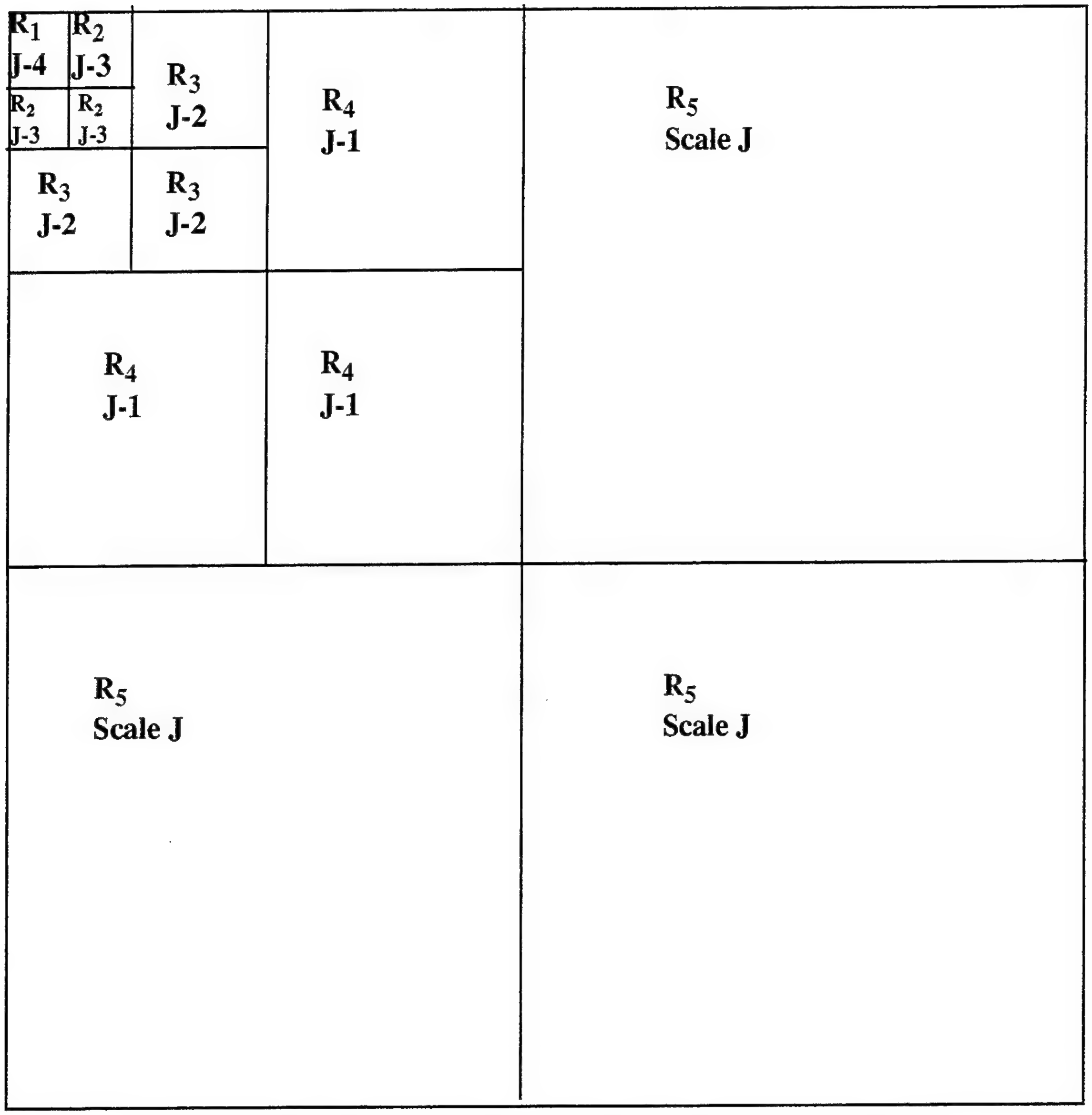


FIGURE 15. Multiresolution Image Compression Using Wavelet Transform

In this scheme, we decompose the signal at five different levels of resolution, which are labeled R_1, R_2, R_3, R_4, and R_5. The feature vector for R_1 consists of the wavelet coefficients at scale J-4, R_2 consists of R_1 enhanced by the wavelet coefficients at scale J-3, etc. The feature vector for R_5 is the entire wavelet decomposition of the test image, as

shown in Figure 15. The data compression ratio C_i for each resolution level R_i is a function of the resolution level used [Ref 9, p. 27]. These ratios are obtained as a ratio of the total bits transmitted to the total number of bits in the original image. Table 1 shows the compression ratios for each of the resolution levels.

Table 1: COMPRESSION RATIOS FOR RESOLUTION LEVELS

Resolution Level	C_i
R_1	256:1
R_2	64:1
R_3	16:1
R_4	4:1
R_5	1:1

G. MULTIRESOLUTION IMAGE RECOGNITION SCHEME

As stated in section III.A, in order to identify rapidly and accurately images of particular interest, we require a way to perform image recognition on images at various resolution levels. One method is to interpolate the low resolution image until it is the same size as the higher resolution image. One can then input this image to the recognition algorithm proposed in Chapter II.

An alternative method is to perform the comparison of the two images in the wavelet domain. In this technique, the distance between the feature vector at resolution level R_1-R_5 and a reference vector is computed. The reference vector is obtained by taking the wavelet transform of the reference image and discarding wavelet coefficients at resolution levels higher than the test image. A block diagram of this scheme is shown in Figure 16.

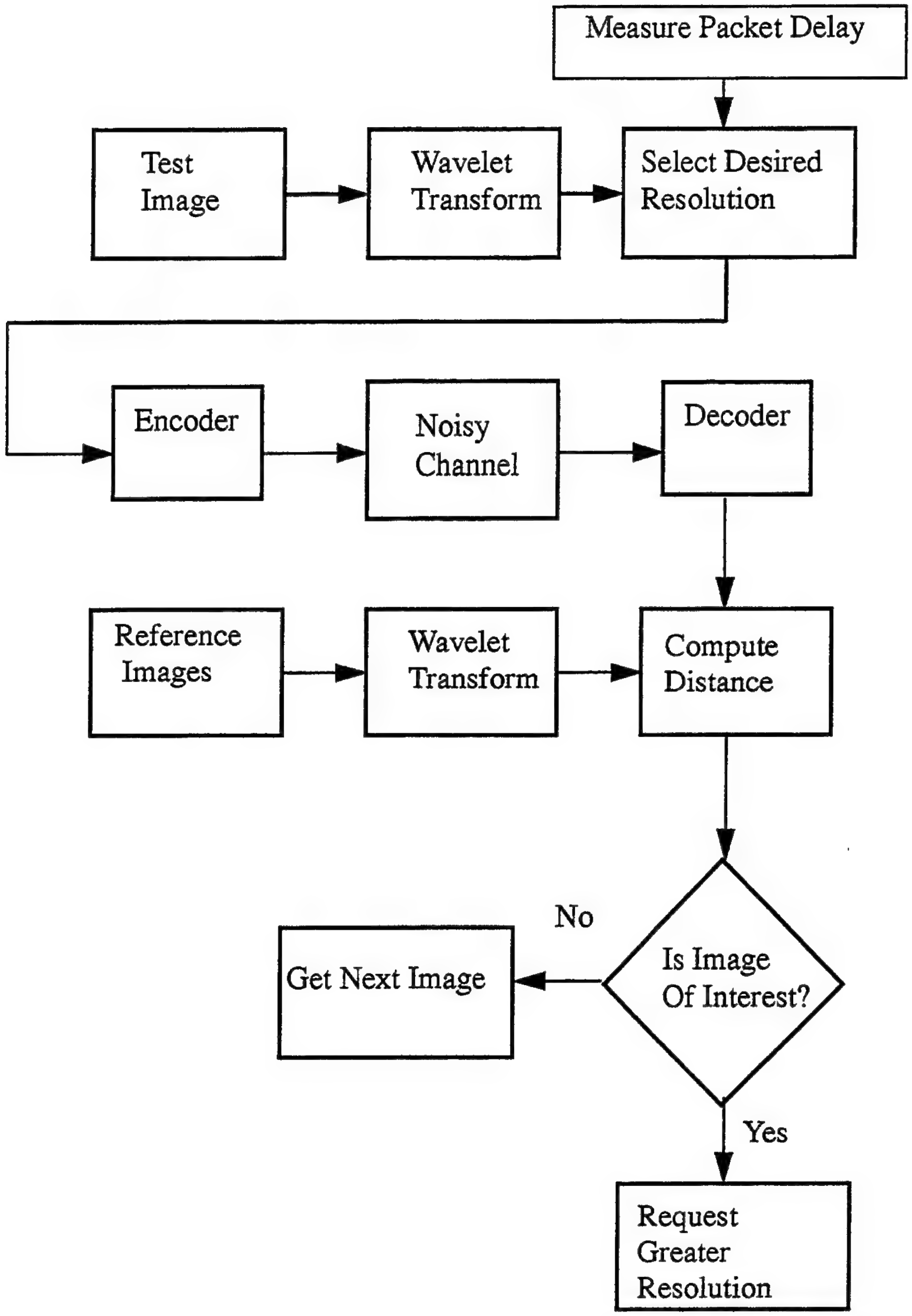


FIGURE 16. Multiresolution Image Compression and Recognition Scheme

The scheme assumes that the user desires to transmit an image which he knows can be associated with images stored in a reference set. In this thesis, the images to be

transmitted are one of 36 alphanumeric characters. Before the user transmits the image, the total packet delay through the transmission channel is measured to determine the current load on the channel. Based on this measurement, the user selects a desired resolution level $R_1 \ldots R_5$ for the image to be transmitted. The wavelet transform of the test image is computed and wavelet coefficients at scales higher than the desired level of resolution are discarded. The image is then encoded using 8 or 12 bits per wavelet coefficient for transmission in the noisy channel. At the receiving end, the received message is decoded and the received wavelet coefficients are formed into a vector. The distance between the received vector is compared with a set of 36 reference vectors consisting of the wavelet coefficients of the reference images. The received vector is associated with the vector with the smallest Euclidean distance. Once this association has been made, the user can determine whether the image is of sufficient interest to request transmission of the remaining wavelet coefficients.

H. SUMMARY

The wavelet transform decomposes the signal into orthogonal components at different scales, which can be used to form feature vectors for image recognition. The signal energy in each subband of the wavelet transform can also be used to create feature vectors. In the next chapter, we use show the performance of a multiresolution image compression and recognition scheme in the presence of channel noise.

IV. IMPLEMENTATION OF MULTIRESOLUTION CHARACTER RECOGNITION

In this chapter we shall implement the proposed multiresolution image recognition scheme and present the results. Performance of the proposed algorithm using wavelet transform coefficients at all five resolution levels as elements of the feature vectors (see Chapter III) is presented first. The use of signal energy in each subband of the wavelet transform as elements of the feature vector is presented next. The use of Fourier transform coefficients as elements of the feature vectors to perform recognition on images which are linearly displaced with respect to the reference image without first registering the test image is then presented. Finally, we show results for each of these cases in the presence of channel noise.

The use of text characters as image objects was chosen for several reasons. First, text characters are easy to segment, so we did not have to develop complicated low-level processing routines which did not relate to the main topic of the thesis. Second, since many text characters look very similar -- a 0 looks much like an O, an E looks much like an F, etc., text characters provide a rigorous test of image recognition performance. Finally, realistic data sets of text characters are easy to generate.

A. IMPLEMENTATION

We shall begin by describing implementation details for each block in the image recognition scheme proposed in Figure 2. The 36 images of text characters we shall use in this thesis were obtained by typing the capital letters A-Z and numbers 0-9 using the OCR-A font in a standard word processing program. The letters were then printed using a standard laser printer and scanned using a flatbed scanner and a commercially available image processing program. To convert the scanned image into a form suitable for processing in Matlab, the scanner output was first saved in JPEG format, then converted to TIFF format, and finally converted to a 64 level grayscale image in Matlab. This character set is shown Figure 17.

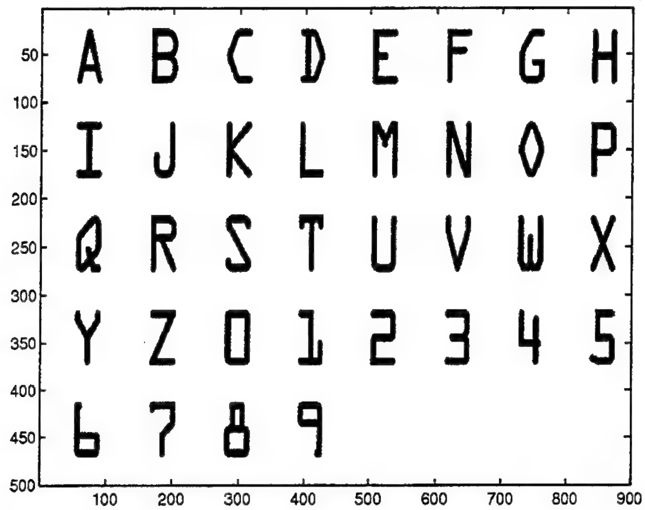


FIGURE 17. Reference Character Set

The scanned letters in Figure 17 have sharp edges, and there is little noise to hinder the recognition process. We tested the effect of a 3x3 median filter for enhancement, but only 1-3 pixels changed value during the filtering process, so we judged this to be insignificant. Also considered were edge sharpening filters, but it was determined that this might create unintended artifacts and would not produce discernible enhancement with such high quality images.

The image in Figure 17 was fed to a segmentation routine which first identified each line of text by summing across the rows of the image and applying a threshold. If the sum was less than the threshold, the algorithm concluded the row consisted entirely of background; conversely, if the sum exceeded the threshold, the row contained part of a character. The threshold was determined based on a histogram of the pixel values. The rows identified as containing text characters were then grouped together; if there was a large break between identified rows, the algorithm concluded that one line of text had ended and another begun.

Next, the individual characters were parsed from each row by applying another threshold. Plots of individual characters obtained from the character set in Figure 17 are shown in Figure 18. The computer code used to generate these characters, **cutter.m**, is

given in the Appendix. These characters form the set of reference images for the recognition system for all trials in this thesis.

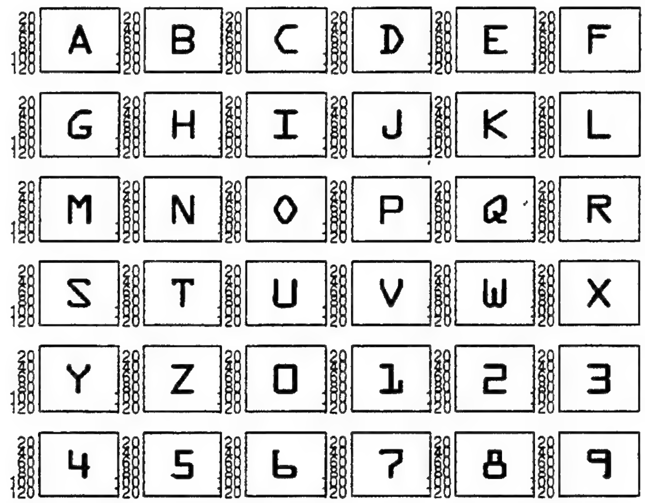


FIGURE 18. Reference Character Set After Segmentation

B. MULTIRESOLUTION RECOGNITION WITH ZERO NOISE ADDED

1. Wavelet Coefficients as Feature Vector Elements

To test the proposed multiresolution recognition scheme, a set of grayscale test characters was created by separately scanning and segmenting several lines of text using the same procedure used to create the reference set. This set of test characters is shown in Figure 19. Note that each alphanumeric character in the reference set is represented in the test set.

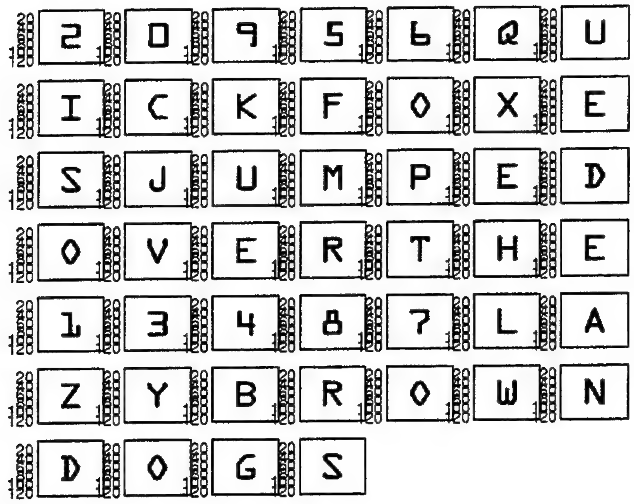


FIGURE 19. Test Character Set

The wavelet transform of each test image and each reference image was computed and the coefficients placed into a feature vector as described in Chapter III. Note that the size of both the test and the reference feature vectors are different for each level of resolution R_1-R_5. The Euclidean distance was computed between the test vector and the reference vector for each image in the reference set. The test character was associated with the text character in the reference set with the smallest Euclidean distance measure. The declared associations were compared with the correct associations to determine recognition performance. We ran this test for the Daubechies filter family of length 4, 12, and 20; each trial yielded 1.000 (100%) recognition performance.

2. Subband Energy as Feature Vector Elements

An alternative method which works with even fewer bits transmitted uses the energy in each subband of the wavelet transform as elements of the feature vector. As given by Equation 20, the energy in a subband is the squared sum of wavelet coefficients in that subband. To compare images at different levels of resolution, we ignore the energy in small-scale subbands. For example, if the test image is transmitted at resolution level R_1, we ignore the energy in subbands for the three smallest scales. Table 2 shows the size of

the wavelet coefficient matrix for the compressed image at each level of resolution and the length of the feature vector for 128 x 128 images such as those used in this thesis.

TABLE 2: FEATURE VECTOR LENGTH USING SUBBAND ENERGY

Resolution Level	Size of Wavelet Coefficient Matrix	Length of Feature Vector
R_1	8 x 8	9
R_2	16 x 16	12
R_3	32 x 32	15
R_4	64 x 64	18
R_5	128 x 128	21

For a length 20 Daubechies filter with zeros channel noise added, we observed 0.9565 recognition performance at resolution level R_1, 0.9783 for R_2, and 1.000 recognition performance for resolution levels R_3-R_5.

3. Fourier Transform Coefficients as Elements of the Feature Vectors

As discussed in Chapter II, one problem which complicates the image recognition problem is classifying images which are not in the same spatial location as the reference images. When using spatial domain features, it is necessary first to register the image to compensate for any translation. An alternative approach is to take the Fourier transform of the image and to use the magnitude of the coefficients as features, which allows one to perform image recognition without first registering the test image.

To test the performance of this method with zero sensor noise, a set of test images was created from the reference set by translating the reference character away from the center by an arbitrary amount. This test set is shown in Figure 20.

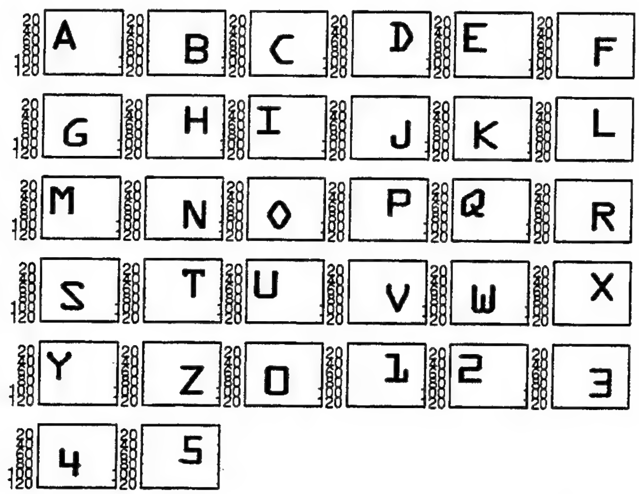


FIGURE 20. Translated Reference Character Set

The two-dimensional Fourier transform of each image in this test and the reference set was computed. The magnitudes of the transform coefficients for each set of characters were formed into feature vectors. The measured distance between the feature vectors for the reference and test sets was zero within the limits of finite computer precision (on the order of 10^{-30}).

To test whether it is possible to perform image recognition without registering the image in the presence of sensor noise, we translated each image in the character set shown in Figure 19 by a random amount. This test set is shown in Figure 21. This trial yielded 0.9565 recognition performance at resolution level R_5. The algorithm incorrectly identified a “6” as a “9” and vice versa. Note that the “6” and “9” in the reference set are nearly identical except for a 180 degree rotation. The measured recognition performance was 0.9130 at resolution level R_3, 0.8261 at resolution level R_2, and 0.3478 at resolution level R_1.

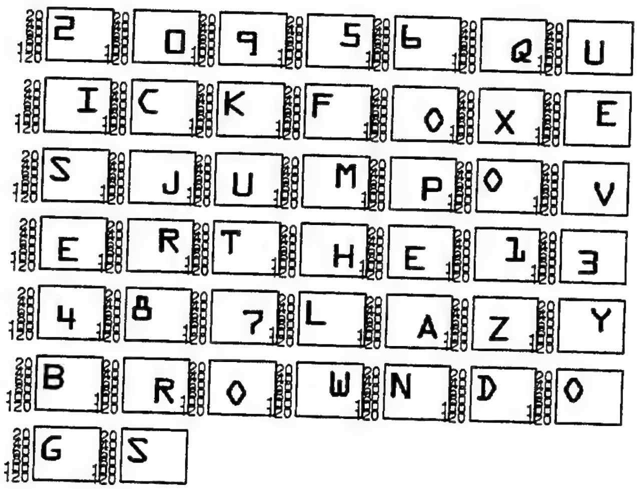


FIGURE 21. Translated Character Test Set

C. MULTIRESOLUTION RECOGNITION WITH CHANNEL NOISE ADDED

1. Wavelet Coefficients as Elements of Feature Vectors for Grayscale Images

To test the performance of the proposed multiresolution image recognition scheme under realistic conditions, it was necessary to simulate the transmission of data over a noisy transmission channel. The feature vectors for the grayscale test images used in the previous section were converted to 8-bit binary data using uniform quantization and Gray coding. To simulate errors due to additive white Gaussian noise in the communications channel, random bit errors were introduced at selected bit error rates (BER) using a random number generator. Twelve trials of the simulation were performed for each filter length, level of resolution, and bit error rate; results presented are obtained by averaging the results for these twelve trials. Figure 22 shows recognition performance versus bit error rate for resolution levels R_1, R_2, R_3, and R_5 for a length 4 Daubechies wavelet filter. The computer code for this trial, **wavetest.m**, is given in the Appendix. Results for resolution level R_4 was similar to that for R_5 and is excluded for the sake of clarity. Figures 23 and 24 show results for length 12 and length 20 Daubechies wavelet filters, respectively. With respect to the

decrease in recognition performance with increasing compression ratios, the performance shown in Figures 23 and 24 is consistent with that shown in Figure 22.

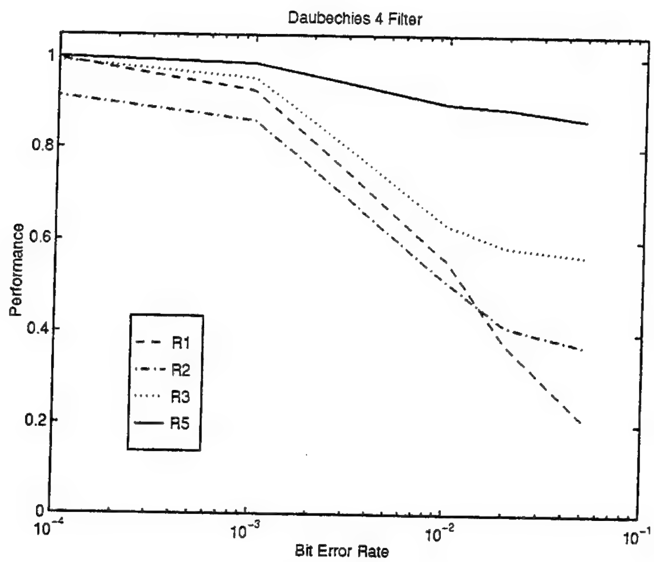


FIGURE 22. Performance with Length 4 Daubechies Filter on Grayscale Image

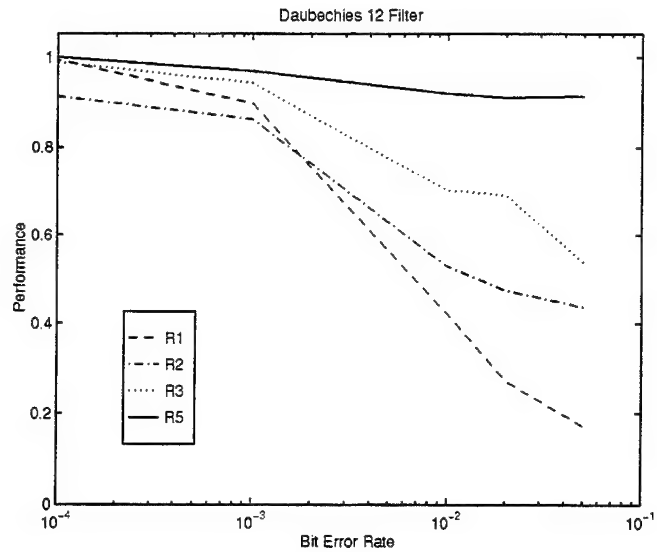


FIGURE 23. Performance with Length 12 Daubechies Filter on Grayscale Image

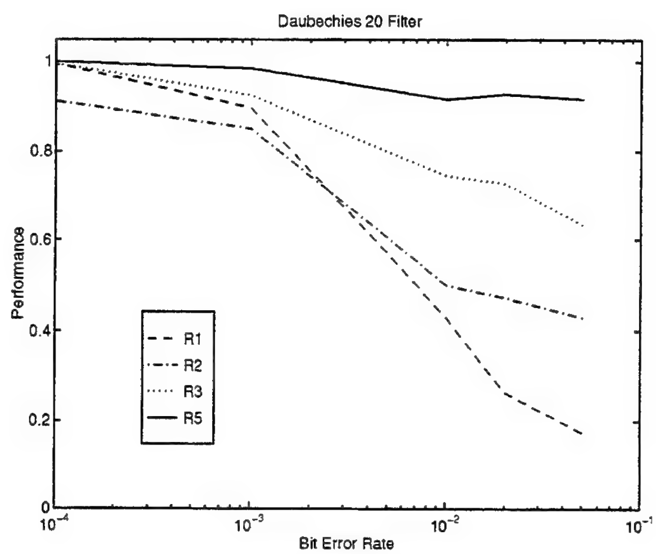


FIGURE 24. Performance with Length 20 Daubechies Filter on Grayscale Image

Recognition performance degrades rapidly when the bit error rate exceeds 10^{-3}. Note that the recognition performance falls with higher compression ratios. This agrees

with our intuition because at higher compression ratios, each bit carries a higher percentage of the information in the image.

2. Wavelet Coefficients as Elements of Feature Vectors for Binary Images

To show the performance of a multiresolution image recognition system for binary images, we tested the multiresolution algorithm using a set of images which had been converted to binary images by histogram thresholding. Figures 25-27 show the results for length 4, 12, and 20 Daubechies filters at R_1, R_2, R_3, and R_5 for grayscale images of the text characters shown in Figure 19 coded using 8 bits per coefficient. Again, the results for resolution level R_4 were similar to those for resolution level R_5 and are omitted for reasons of clarity. The recognition performance for binary images is superior to that for grayscale images and less sensitive to bit errors. We observe that the feature vectors for binary images are more distinct than the feature vectors created using grayscale images. Therefore, we conclude that if the prominent features in an image depend primarily on the shape of the object, as in this thesis, binary thresholding may be a better approach.

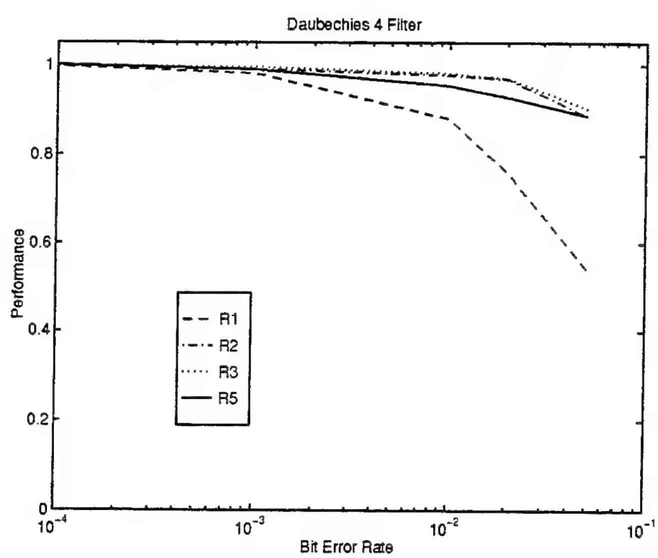


FIGURE 25. Performance with Length 4 Daubechies Filter on Binary Image

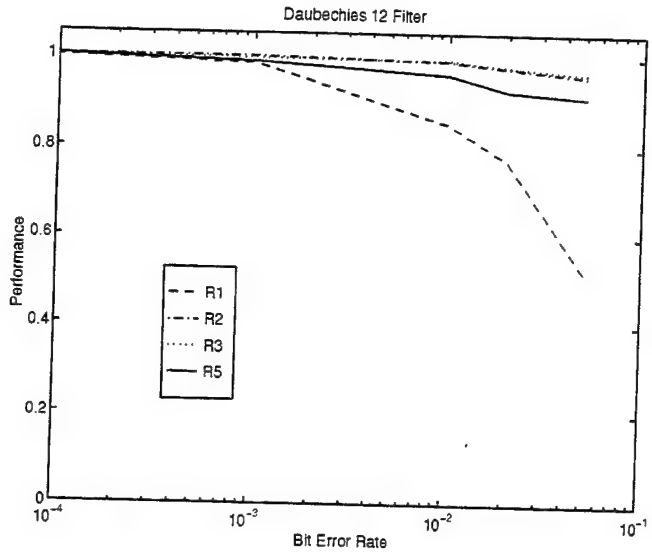


FIGURE 26. Performance with Length 12 Daubechies Filter on Binary Image

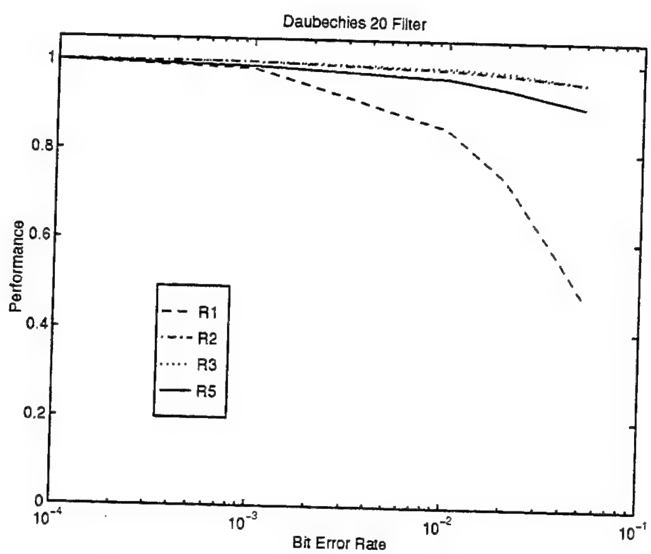


FIGURE 27. Performance with Length 20 Daubechies Filter on Binary Image

3. Recognition Performance as a Function of Filter Length

We also observed that, in general, longer filter lengths yielded better performance for a given bit error rate at a given level of resolution for both binary and grayscale images. Longer filter lengths give better performance because there is less aliasing in the subbands due to the sharper rolloff in frequency response near the cutoff frequency, as shown in Figure 28. Less aliasing results in less spillover of signal content from one subband into another, thus driving the feature vectors further apart.

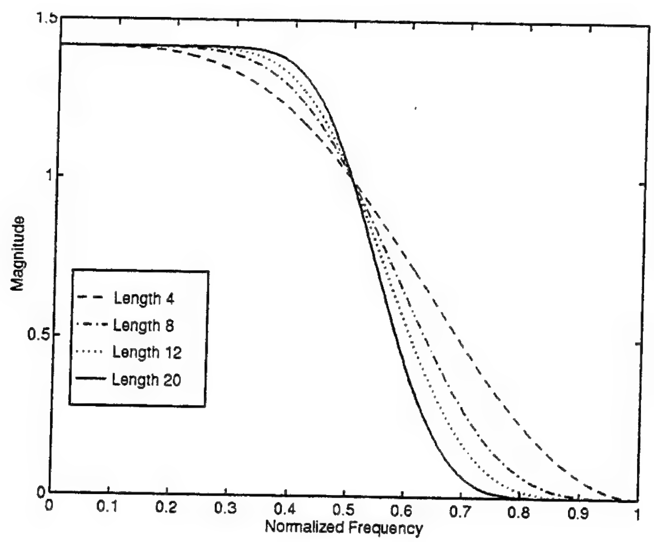


FIGURE 28. Frequency Response Plots for Filters of Length 4, 8, 12, and 20

Figures 29-31 show performance vs. BER for R_1, R_2, and R_3 for length 4, 12, and 20 Daubechies wavelet filters; R_4 and R_5 both showed results near 1.000 for each filter length.

One factor which counters the increase in performance with longer filter lengths is the effect of the length of support for the signal. If a signal is zero outside of some range between points A and B, the distance AB is said to be the support of the signal. The output of a wavelet filter tends to have a sharp response when the support of the signal is as long as the support of the filter. This is similar to the matched filter effect. Note that in Figure 29, for which the signals are most compressed and thus have the shortest support, the length

4 Daubechies wavelet filter outperforms the longer filters. In Figures 30 and 31, this effect is less prominent.

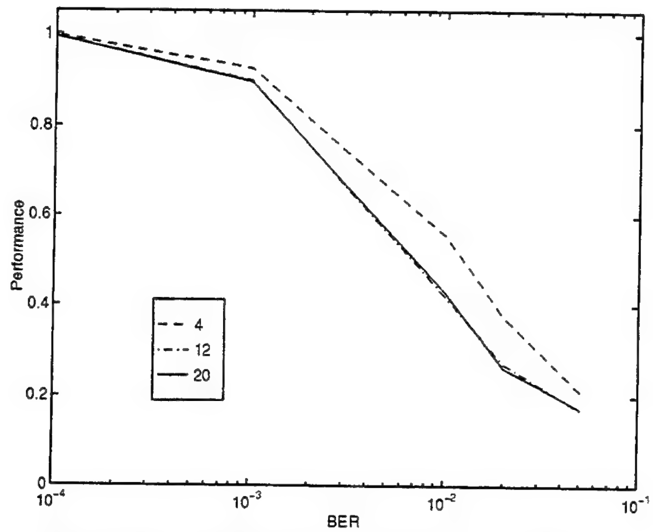


FIGURE 29. Performance for Daubechies Wavelet Filters of Length 4, 12, and 20 at Resolution Level R1

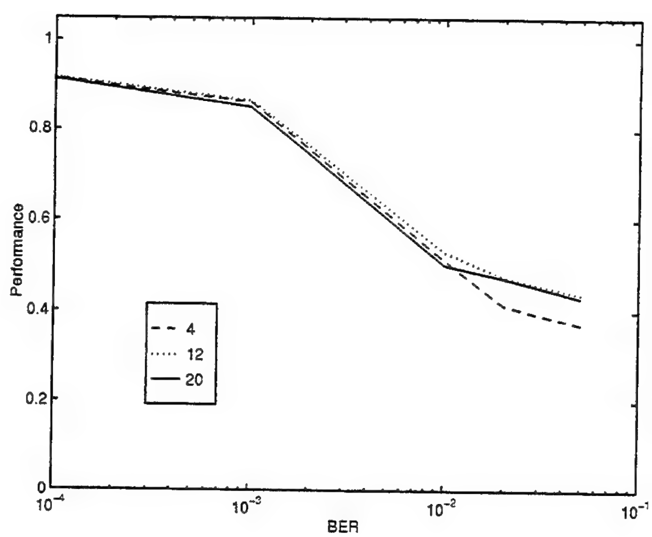


FIGURE 30. Performance for Daubechies Wavelet Filters of Length 4, 12, and 20 at Resolution Level R2

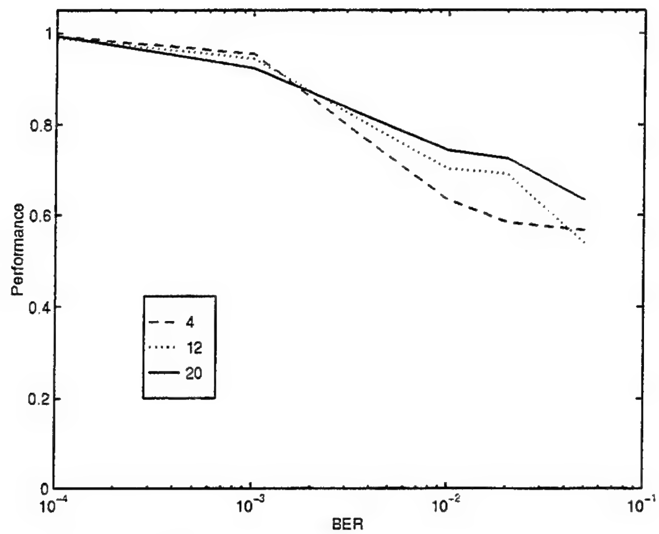


FIGURE 31. Performance for Daubechies Wavelet Filters of Length 4, 12, and 20 at Resolution Level R3

4. Recognition Performance as a Function of Bit Resolution

To assess the effect of bit resolution in the performance of the multiresolution scheme, we repeated the procedure discussed in the previous section using 12 bits per coefficient. Due to very large computational and memory requirements for this test, we conducted only one trial for each BER and filter length. Figures 32-34 (single trial) show that recognition performance is more robust in the presence of channel noise with 12 bit coding as compared to the performance shown in Figures 25-27 using 8 bit coding (averaged over 12 trials).

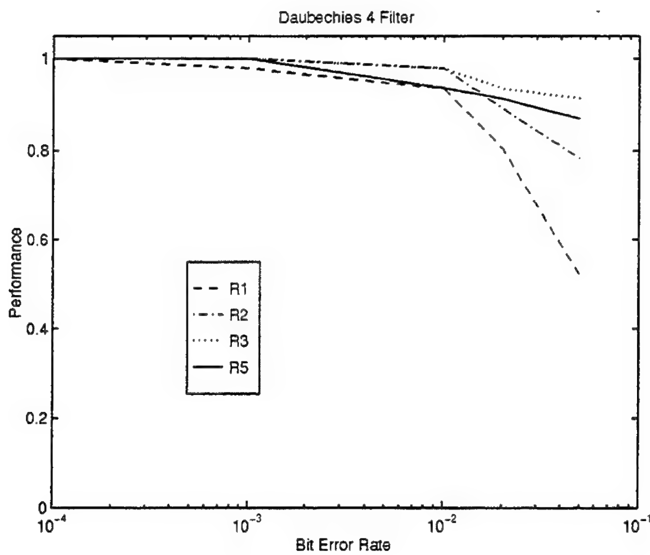


FIGURE 32. Performance for Daubechies 4 Filter at Resolution Levels R1, R2, R3, and R5 with 12 bit coding

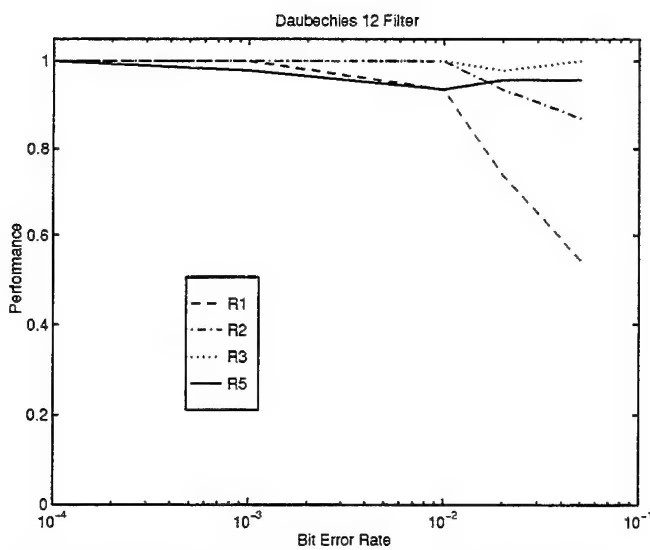


FIGURE 33. Performance for Daubechies 12 Filter at Resolution Levels R1, R2, R3, and R5 with 12 bit coding

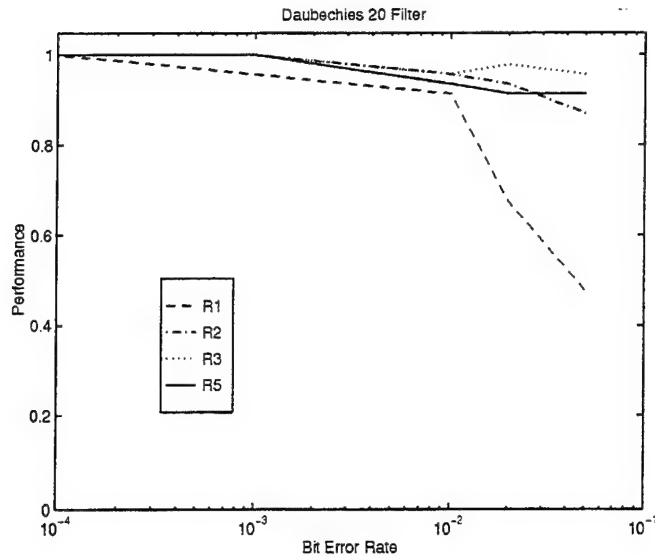


FIGURE 34. Performance for Daubechies 20 Filter at Resolution Levels R1, R2, R3, and R5

5. Subband Energy as Elements of the Feature Vectors

Figures 35-37 show the performance for multiresolution image recognition using subband energy as feature vectors in the presence of channel noise. Computer code for this method, **bandtest.m,** is given in the Appendix.

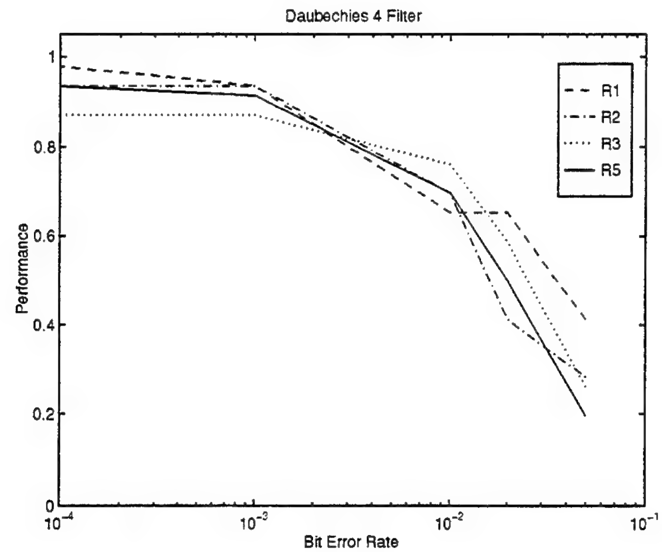


FIGURE 35. Perfomance of a Length 4 Daubechies Filter Using Subband Energy As Features

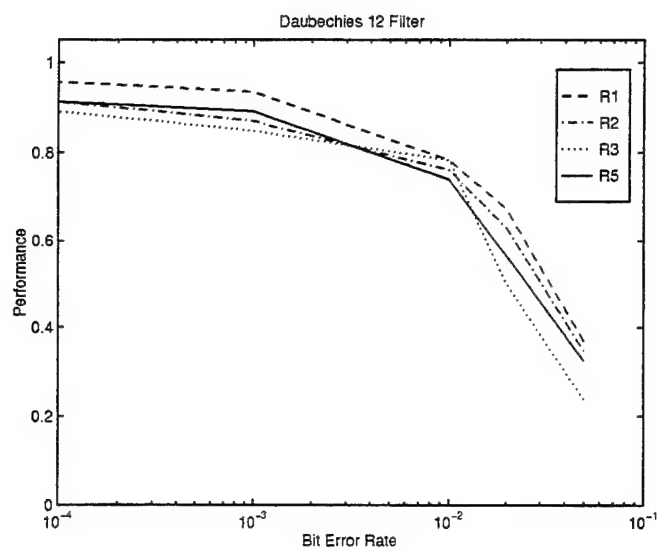


FIGURE 36. Perfomance of a Length 12 Daubechies Filter Using Subband Energy As Features

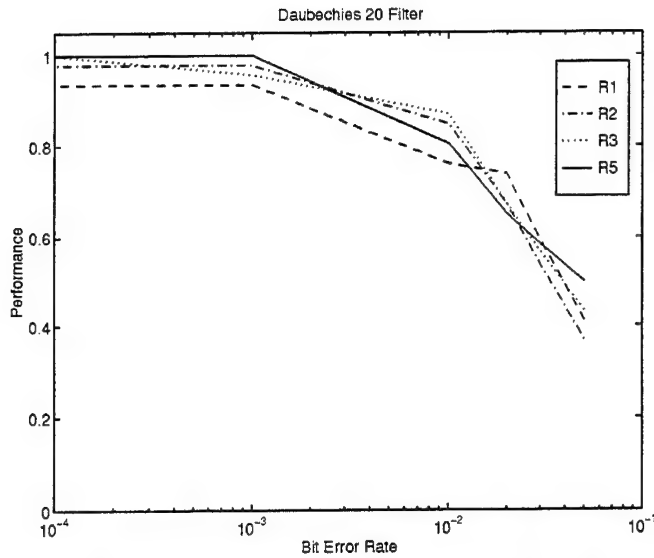


FIGURE 37. Perfomance of a Length 20 Daubechies Filter Using Subband Energy As Features

Recognition performance using this method decreases rapidly in the presence of channel noise. Since each feature vector element represents a higher proportion of the total information transmitted, it is reasonable that bit errors introduced by the channel would have a greater effect on the recognition performance than when using the wavelet coefficients as feature vectors.

6. Fourier Transform Coefficients as Elements of the Feature Vectors

Combining translation invariant recognition with multiresolution analysis, we first computed the wavelet transform of the image and then compute the two-dimensional Fourier transform of the wavelet coefficients in each subband used at the desired level of resolution. For example, for resolution level R_1, we decomposed the image using the wavelet transform and then computed the two dimensional Fourier transform of each subband of the 8 x 8 matrix of coefficients used in R_1. The magnitudes of the resulting Fourier coefficients were formed into a test vector for transmission. Computer code for this trial, **transtest.m** is presented in the Appendix. Figures 38-41 show recognition

performance for this technique in the presence of channel noise. Performance using this technique is lower than the recognition performance for images which have not undergone translation. As in the case with no noise, the results for resolution level R_1 are much worse than the performance for images at higher levels of resolution.

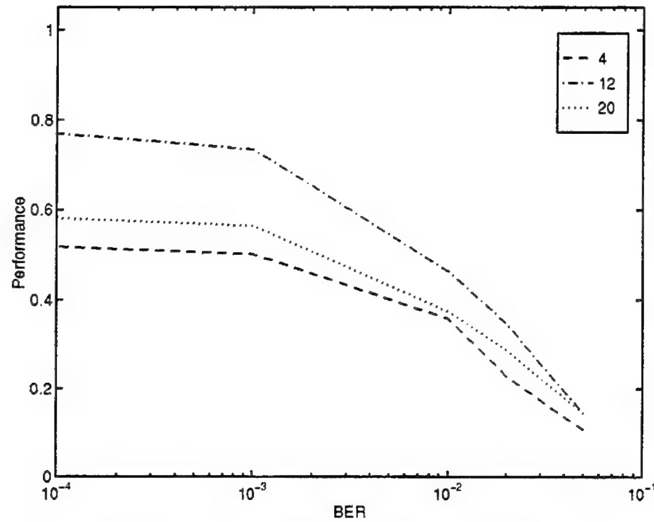


FIGURE 38. Performance for Daubechies Filters of Length 4, 12, and 20 at Resolution Level R1 with Linear Translation

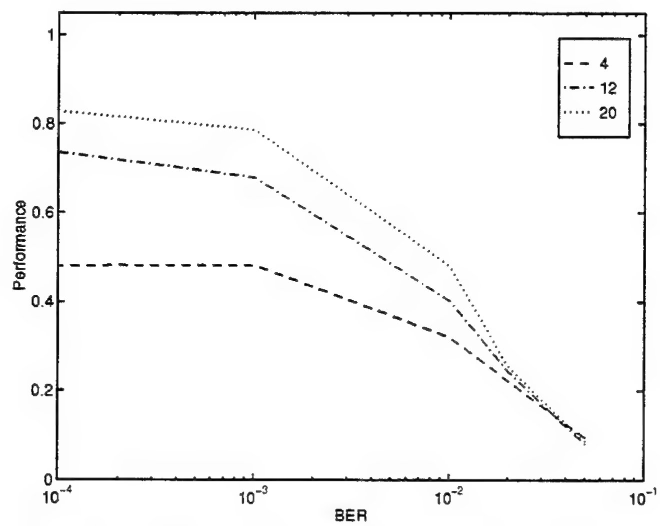


FIGURE 39. Performance for Daubechies Filters of Length 4, 12, and 20 at Resolution Level R2 with Linear Translation

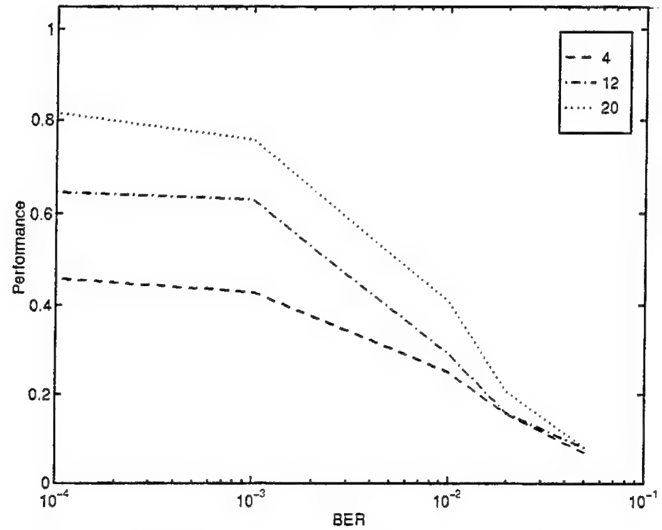


FIGURE 40. Performance for Daubechies Filters of Length 4, 12, and 20 at Resolution Level R3 with Linear Translation

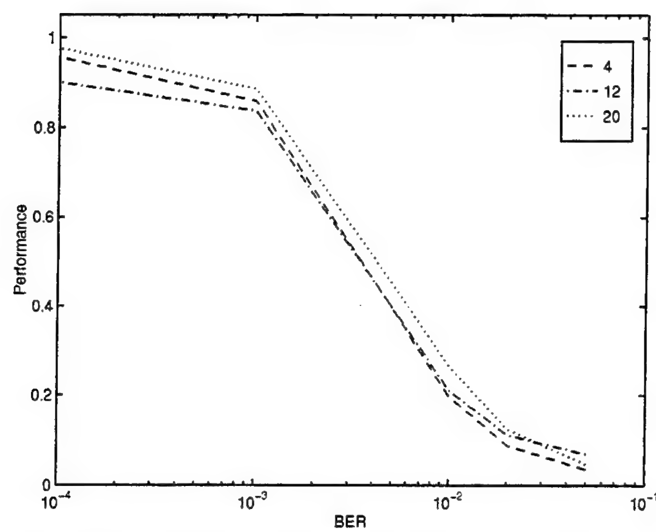


FIGURE 41. Performance for Daubechies Filters of Length 4, 12, and 20 at Resolution Level R5 with Linear Translation

D. MULTIRESOLUTION RECOGNITION WITH AIRCRAFT LINE IMAGES

To show that multiresolution image recognition can be used as part of an automatic target recognition scheme, we created the following test set by scanning clip art images from a standard image processing application. This test set is shown in Figure 42.

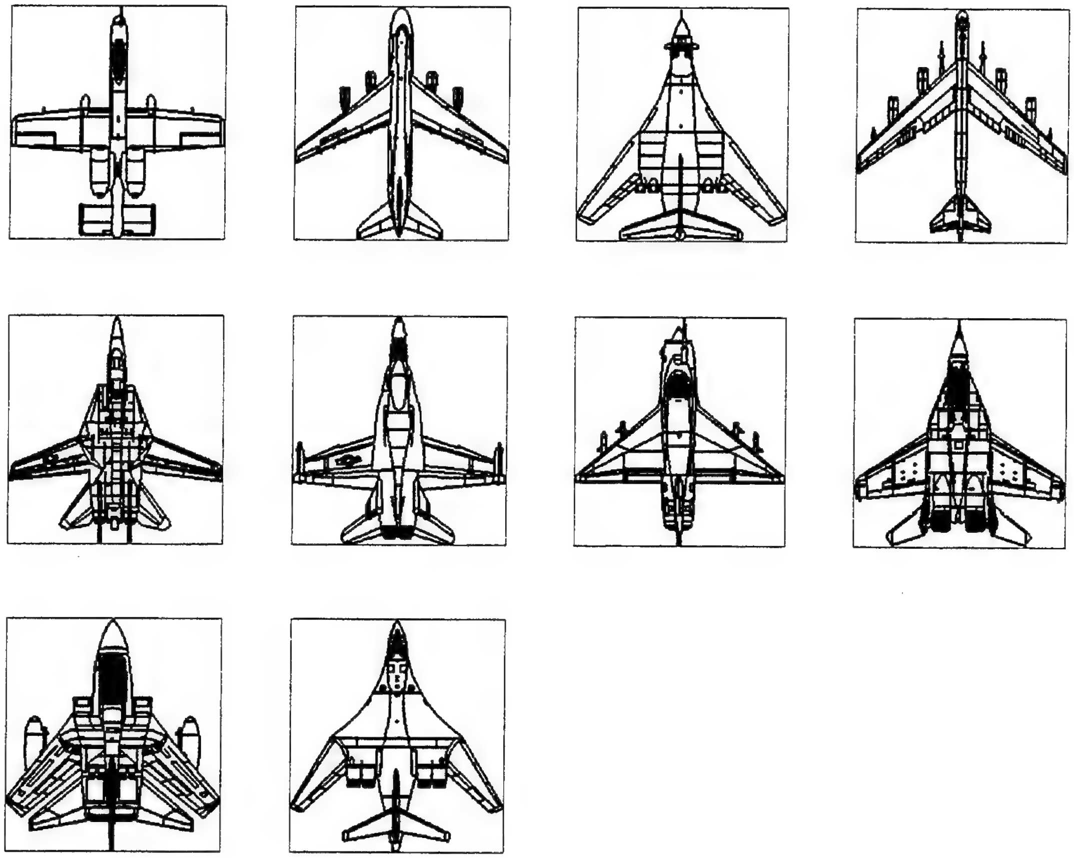


FIGURE 42. Aircraft Line Drawing Test Set

Using the test set shown in Figure 42 and a separately scanned reference set, we tested multiresolution image recognition performance with wavelet coefficients as feature vectors. The computer code for this trial, **planetest.m**, is given in the Appendix. With no channel noise added, the scheme produced 1.000 recognition performance at all resolution levels and for each filter length. With channel noise added to simulate random errors caused by additive white Gaussian noise in the transmission channel, the scheme produced 1.000 recognition performance.at resolution levels R_2, R_3, and R_4 for each filter length. At resolution level R_1, we obtained the performance shown in Figure 43.

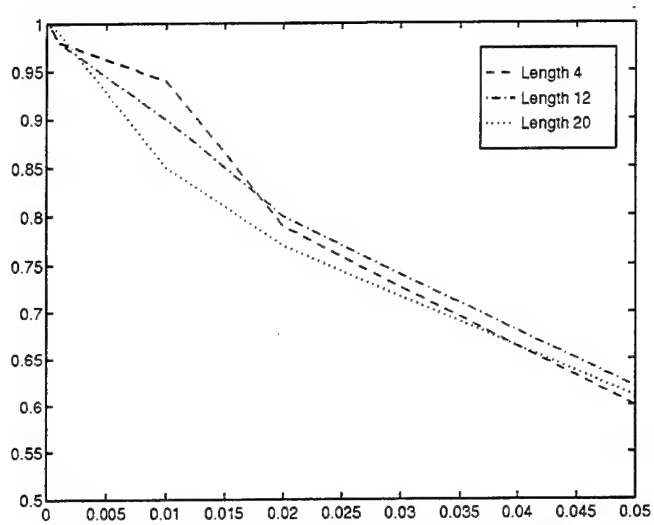


FIGURE 43. Recogntion Performance vs. BER for Aircraft Test Set for Compression Ration of 256:1

Again, the degradation in recognition performance for high compression ratios with increasing channel noise results from the fact that each bit carries a higher percentage of the total image information. The recognition performance at resolution levels R_2-R_5 in this example is superior to that obtained for the text characters for two reasons. First, the line drawings are more distinct than the text characters, which means that the feature vectors obtained from them are further apart. Second, the size of the reference set is smaller, reducing the chance of erroneous identification.

E. SUMMARY

The wavelet transform can be used to create feature vectors to perform multiresolution image recognition on grayscale and binary images. Recognition performance for different resolution levels, bit error rates, and filter lengths for grayscale and binary images is summarized in Tables 3 and 4.

Table 3: MULTIRESOLUTION IMAGE RECOGNITION PERFORMANCE FOR GRAYSCALE IMAGES USING WAVELET COEFFICIENTS FOR FEATURE VECTORS

	Filter Length	R_1	R_2	R_3	R_5
C_i		256:1	64:1	16:1	1:1
BER = 0.05	4	0.2120	0.3714	0.5670	0.8636
	12	0.1712	0.4366	0.5390	0.9150
	20	0.1739	0.4293	0.6341	0.9150
BER = 0.02	4	0.3750	0.4130	0.5851	0.8874
	12	0.2717	0.4746	0.6920	0.9130
	20	0.2636	0.4728	0.7264	0.9269
BER = 10^{-2}	4	0.5571	0.5109	0.6341	0.8972
	12	0.4212	0.5290	0.7029	0.9209
	20	0.4293	0.5000	0.7446	0.9150
BER = 10^{-3}	4	0.9266	0.8605	0.9547	0.9881
	12	0.8995	0.8641	0.9438	0.9704
	20	0.8967	0.8496	0.9239	0.9842
BER = 10^{-4}	4	0.9973	0.9130	0.9928	1.0000
	12	0.9946	0.9149	0.9891	1.0000
	20	0.9946	0.9112	0.9946	1.0000

Table 4: MULTIRESOLUTION IMAGE RECOGNITION PERFORMANCE FOR BINARY IMAGES USING WAVELET COEFFICIENTS FOR FEATURE VECTORS

	Filter Length	R_1	R_2	R_3	R_5
C_i		256:1	64:1	16:1	1:1
BER = 0.05	4	0.5417	0.8870	0.9040	0.8877
	12	0.5217	0.9609	0.9656	0.9149
	20	0.4819	0.9609	0.9583	0.9040
BER = 0.02	4	0.7609	0.9696	0.9710	0.9293
	12	0.7717	0.9783	0.9801	0.9275
	20	0.7391	0.9783	0.9837	0.9438
BER = 10^{-2}	4	0.8804	0.9783	0.9819	0.9547
	12	0.8533	0.9913	0.9928	0.9620
	20	0.8533	0.9870	0.9909	0.9674
BER = 10^{-3}	4	0.9819	0.9913	0.9964	0.9909
	12	0.9855	1.0000	0.9964	0.9891
	20	0.9855	1.0000	1.0000	0.9909
BER = 10^{-4}	4	0.9982	1.0000	0.9982	1.0000
	12	0.9982	1.0000	1.0000	1.0000
	20	1.0000	1.0000	1.0000	1.0000

If the prominent features of an image depend on shape, as in the images used in this thesis, binary thresholding is an important preprocessing step. Recognition performance varies with the bit error rate. Longer filter lengths generally provide better performance, but as the support of the signal decreases with increasing compression, shorter filter lengths can provide better results. Performance degrades as the compression ratio increases, but it is possible to achieve 1.000 recognition performance if the probability of bit error is less than 10^{-3}. If Fourier transform coefficients are used as elements of the feature vectors, it is possible to perform multiresolution image recognition of images which have undergone

linear translation without registering the image. Performance degradation for compression ratios higher than 64:1 is particularly severe. Multiresolution image recognition can be extended to images other than text characters and could be used as part of an automatic target recognition system.

V. CONCLUSIONS

A scheme for multiresolution image recognition was presented. The scheme allows a user to adjust the flow of data over a digital communications channel by varying the desired resolution of the transmitted image. The transmission of grayscale text character images at several levels of resolution over a noisy digital channel was simulated, and plots of the recognition performance versus bit error rate were presented. Fourier coefficients were used to perform multiresolution image recognition on images which had undergone linear translation without first registering the image. An extension to the use of binary text characters and to aircraft line drawings was made.

Recognition performance degraded as bit error rate increased and as the compression ratio increased. Longer filter lengths provided better performance at higher levels of resolution, but shorter filter lengths performed better at a compression ratio of 256:1. Results for data coded at 12 bits per coefficient were superior to those for data coded at 8 bits per coefficient. Performance for binary images exceeded that for grayscale images, indicating that conversion to binary via histogram thresholding may be a better approach for recognizing images whose features depend primarily on shape. Results for aircraft line drawings were superior to those for either the grayscale or binary text characters.

This scheme is useful in any situation in which image recognition is critical and the available bandwidth is limited and subject to change according to the traffic on the channel. Typical applications include the recognition of images transmitted from a remote sensing platform and searching an image archive at a remote site.

Future research is needed to extend the proposed approach to different types of images, such as real images of military targets, medical images, and human faces. Recognition performance in the presence of sensor noise, image rotation, and image aspect variation are other important areas of research.

APPENDIX A

Matlab code implementing the multiresolution image recognition algorithms presented in this thesis is provided below.

aconv.m

```
function y = aconv(f,x)
% aconv -- Convolution Tool for Two-Scale Transform
% Usage
%   y = aconv(f,x)
% Inputs
%   f   filter
%   x   1-d signal
% Outputs
%   y   filtered result
%
% Description
%   Filtering by periodic convolution of x with the
%   time-reverse of f.
%
% See Also
%   iconv, UpDyadHi, UpDyadLo, DownDyadHi, DownDyadLo
%

n = length(x);
p = length(f);
if p < n,
  xpadded = [x x(1:p)];
else
  z = zeros(1,p);
  for i=1:p,
  imod = 1 + rem(i-1,n);
  z(i) = x(imod);
  end
  xpadded = [x z];
end
fflip = reverse(f);
ypadded = filter(fflip,1,xpadded);
y = ypadded(p:(n+p-1));

%
% Copyright (c) 1993. David L. Donoho
%


%
% Part of WaveLab Version .700
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
```

```
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

bandtest.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Determine performance of multiresolution image recognition system
%using subband energy as feature vectors
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all;
loadgrayref;%Load reference images
loadgraytest;%Load test images
colnum = 6;%Initialize global variables for simulation
global START_OK;
START_OK = 1;

Pgrayband1 = zeros(5,3);%Initialize output variable - Overall correct recocnition
for trials = 1:12;%Run simulation for 12 trials
Pcorr = zeros(5,3);%Initialize intermediate output variables
m = zeros(46,36);%Initialize Euclidean distance measure matrix
b = [4 12 20];%Run for 3 filter lengths
biterr = [.05 .02 .01 .001 .0001];%Run for 5 bit error probabilities

%%%%%%%%
%Main program
%Perform simulated transmission over noisy channel for each filter length and
%bit error probability for this level of resolution (R1)

for u2 = 1:length(biterr);%Run for all bit error probabilities
thisbiterr = biterr(u2);%P(b) for this trial
for u = 1:length(b);%Run for all filter lengths
count2 = 1;%test image index
y = zeros(1,46);%store identified text character in this vector
qmf = MakeONFilter('Daubechies',b(u));%Create daubechies filter for this trial

for k = 1:5;%Run for all test characters
for j = 1:13;
%load test character
if exist(['graytest' num2str(k) num2str(j)]) == 0;
break;
else   t = eval(['graytest' num2str(k) num2str(j)]);

%Compare t with all reference characters
count = 1;
for p = 1:5;
  for q = 1:8;
    if p == 5 & q > 4 ;
      break;
    else r = eval(['grayref' num2str(p) num2str(q)]);
      r = FWT2_PO(r,1,qmf);%Compute wavelet transform of reference image
      r = bandenergy2(r);%Compute feature vector using subband energy
```

```
            if count == 1;%If this is the first reference image
              t = FWT2_PO(t,1,qmf);%compute wavelet transform of
                    %test image
              t = bandenergy2(t);%Compute feature vector using subband energy
              t = quantize(t,8,'scale');%Quantize uniformly using 8 bits
                    %per coefficient
              t = bin_enc(t,8);%Encode using natural binary code
              t = bin2gray(t);%Convert to gray coding
              %Introduce random errors using random number generator with
              %probability = thisbiterr

              noise = zeros(size(t));
              bitchange = find(rand(size(t)) < thisbiterr);
              noise(bitchange) = ones(size(bitchange));
              %Corrupt transmitted signal t with random bit errors found above
              t = xor(t,noise);
              t = gray2bin(t);%convert to natural binary code
              t = bin_dec(t);%decode signal
            end;
            m(count2,count) = (r-t)*(r-t)';%Compute distance between reference
                    %and test vectors
            count = count + 1;%increment reference vector index
          end;
        end;
      end;
      end;
      [n,j] = min(m(count2,:));%Determine index of reference image with minimum
              %Euclidean distance
      y(count2) = num2let(j);%Convert to ASCII character
      count2 = count2+1;%Increment test vector index
      end;
      end;
      %Compare results obtained this trial with correct answer
      ideal = ['2' '0' '9' '5' '6' 'Q' 'U' 'I' 'C' 'K' 'F' 'O' 'X' 'E' 'S' 'J' 'U' 'M' 'P' 'E' 'D' 'O' 'V' 'E'
'R' 'T' 'H' 'E' '1' '3' '4' '8' '7' 'L' 'A' 'Z' 'Y' 'B' 'R' 'O' 'W' 'N' 'D' 'O' 'G' 'S'];
      Pcorr(u2,u) = 1 - length(find(y - ideal))/length(ideal);%Determine error performance
      end;
      end;
      Pgrayband1 = Pgrayband1 + Pcorr%Determine overall performance as sum of all trials
      save Pgrayband1 Pgrayband1%Store overall performance
      end;
```

cutter.m

```
%%%%%%%%%%%%%%%%%%%%%%%
%Preprocessing
%%%%%%%%%%%%%%%%%%%%%%%
clear all;
load ocralastref;%stored reference image for OCR-A data
lpf = [.25*ones(1,3); .25 1 .25; .25*ones(1,3)]/3;%Low Pass Filter
x = filter2(lpf,x);%Use Low-Pass Filter on Image
x = medfilt2(x,[3 3]);%Use Median Filter on Result
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%
%Identify Lines of Text
%%%%%%%%%%%%%%%%%%%%%%%%%

[m,n] = size(x);
linex = [];%Stores identified lines of text
liney = [];%Stores identified columns of text
y = sum(x');%Sum down rows
linex = find(y > 2.2E5);%Apply threshold
linex2 = [linex(2:length(linex)) 0];
[n,j] = find((linex - linex2) <= -10);%If there is a large break between
%identified rows of text, make a new
%row
j2 = j+1;
linfindxstart = linex(j);%Identifies beginning of row
linfindxend  = linex(j2);%Identifies end of row


%Create a matrix of all identified
for k = 1:[length(linfindxstart)];%lines of text
eval(['set' num2str(k) ' = x([linfindxstart(k):linfindxend(k)],:); ']);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%
%Parse individual characters from lines of text
%%%%%%%%%%%%%%%%%%%%%%%%%%
count = 0; %Total number of Characters
for k = 1:length(linfindxstart);
if count > 46;%Quit when count = 46;
break;
end;
y = sum(eval(['set' num2str(k)]));%Work on the kth row of the original image
liney = find(y > 1.5E4);%Apply Threshold
liney2 = [liney(2:length(liney)) 0];
[n,j] = find((liney - liney2) <= -10);%If there is a sharp break between identifed
  %columns, start a new character
j2 = j+1;
linfindystart = liney(j);%Stores the starting column for character
linfindyend  = liney(j2);%Stores the last column for the character

for v = 1:[length(linfindystart)];%Separate out and store the individual characters
  %in a square matrix of dyadic size
count = count + 1;
t = eval(['set' num2str(k) '(:,(linfindystart(v):linfindyend(v)))']);
eval(['ocralastref' num2str(k) num2str(v) ' = t;']);
  %Determine the largest dyadic number greater than the
  %maximum dimension of the image
[n,Jcol] = dyadlength(eval(['ocralastref' num2str(k) num2str(v) '(1,:)']));
[n,Jrow] = dyadlength(eval(['ocralastref' num2str(k) num2str(v) '(:,1)']));
```

```
Jref = max(Jcol,Jrow);
t = ['ocralastref' num2str(k) num2str(v)];
t = eval(t);
colpad = 2^Jref - size(t,2);%Number of pixels to pad columns
rowpad = 2^Jref - size(t,1);%Number of pixels to pad rows


colpadleft = floor(colpad/2);%Pad rows and columns
colpadright = colpad - colpadleft;
rowpadtop = floor(rowpad/2);
rowpadbot = rowpad - rowpadtop;
eval(['ocralastref' num2str(k) num2str(v) ' = [254*ones(size(t,1),colpadleft) t
254*ones(size(t,1),colpadright)];']);
t = eval(['ocralastref' num2str(k) num2str(v)]);
eval(['ocralastref' num2str(k) num2str(v) ' = [254*ones(rowpadtop,size(t,2)); t;
254*ones(rowpadbot,size(t,2))];']);


t = eval(['ocralastref' num2str(k) num2str(v)]);
t = t/sqrt(sum(sum(t.^2)));%Normalize resulting image to have energy = 1;

eval(['grayref' num2str(k) num2str(v) ' = t;']);%Store the result in its own file
eval(['save grayref' num2str(k) num2str(v) ' grayref' num2str(k) num2str(v)]);
end;


end;
end;

downdyadhi.m
function d = DownDyadHi(x,qmf2)
% DownDyadHi -- Hi-Pass Downsampling operator (periodized)
% Usage
%   d = DownDyadHi(x,f)
% Inputs
%   x   1-d signal at fine scale
%   f   filter
% Outputs
%   y   1-d signal at coarse scale
%
% See Also
%   DownDyadLo, UpDyadHi, UpDyadLo, FWT_PO, iconv
%
d = aconv(qmf2,x);
n = length(d);
d = d(1:2:(n-1));

%
% Copyright (c) 1993. Iain M. Johnstone
%
```

```
%
% Part of WaveLab Version .700
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

downdyadlo.m

```
function d = DownDyadLo(x,qmf1)
% DownDyadLo -- Lo-Pass Downsampling operator (periodized)
% Usage
%   d = DownDyadLo(x,f)
% Inputs
%   x    1-d signal at fine scale
%   f    filter
% Outputs
%   y    1-d signal at coarse scale
%
% See Also
%   DownDyadHi, UpDyadHi, UpDyadLo, FWT_PO, aconv
%
d = aconv(qmf1,x);
n = length(d);
d = d(1:2:(n-1));

%
% Copyright (c) 1993. Iain M. Johnstone
%


%
% Part of WaveLab Version .700
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

dyadlength.m

```
function [n,J] = dyadlength(x)
% dyadlength -- Find length and dyadic length of array
% Usage
%   [n,J] = dyadlength(x)
% Inputs
%   x    array of length n = 2^J (hopefully)
```

```
% Outputs
%   n   length(x)
%   J   least power of two greater than n
%
% Side Effects
%   A warning is issued if n is not a power of 2.
%
% See Also
%   quadlength, dyad, dyad2ix
%
 n = length(x) ;
 J = ceil(log(n)/log(2));
 %if 2^J ~= n ,
  %   disp('Warning in dyadlength: n != 2^J')
 end


%
% Part of WaveLab Version .700
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

fftid2.m

```
%function fftid2: takes the 2 - dimensional Fourier transform of each subband
%of the wavelet coefficient matrix wc and places the normalized Fourier
%coefficient magnitudes into a feature vector

function idvector = fftid2(wc);
idvector = [];%Initialize output vector
[n,J] = quadlength(wc);%Determine dyadic size of square array

for k = J:-1:1;%Perform for each scale
high = (2^(k-1) +1):2^k;%High-pass range
low = 1:2^(k-1);%Low-pass range
N = length(low)^2;%maximum dimension

%For each of the three subbands at this scale, take the 2-
%dimensional Fourier transform and append the magnitude of each onto
%a feature vector
idvector = [idvector reshape(abs(fft2(wc(high,low))),1,N)];
idvector = [idvector reshape(abs(fft2(wc(low,high))),1,N)];
idvector = [idvector reshape(abs(fft2(wc(high,high))),1,N)];
end;
%Add the (1,1) wavelet coefficient
idvector = [idvector wc(1)];
%Normalize the resulting vector
idvector = idvector/length(idvector);
```

```
idvector = idvector/sqrt(sum(idvector.^2));
```

FWT2_PO.m

```
function wc = FWT2_PO(x,L,qmf)
% FWT2_PO -- 2-d MRA wavelet transform (periodized, orthogonal)
% Usage
%   wc = FWT2_PO(x,L,qmf)
% Inputs
%   x     2-d image (n by n array, n dyadic)
%   L     coarse level
%   qmf   quadrature mirror filter
% Outputs
%   wc    2-d wavelet transform
%
% Description
%   A two-dimensional Wavelet Transform is computed for the
%   array x.  To reconstruct, use IWT2_PO.
%
% See Also
%   IWT2_PO, MakeONFilter
%
[n,J] = quadlength(x);
wc = x;
nc = n;
for jscal=J-1:-1:L,
top = (nc/2+1):nc; bot = 1:(nc/2);
for ix=1:nc,
row = wc(ix,1:nc);
wc(ix,bot) = downdyadlo(row,qmf);
wc(ix,top) = downdyadhi(row,qmf);
end
for iy=1:nc,
row = wc(1:nc,iy)';
wc(top,iy) = downdyadhi(row,qmf)';
wc(bot,iy) = downdyadlo(row,qmf)';
 end
nc = nc/2;
end

%
% Copyright (c) 1993. David L. Donoho
%


%
% Part of WaveLab Version .700
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

iconv.m

```
function y = iconv(f,x)
% iconv -- Convolution Tool for Two-Scale Transform
% Usage
%   y = iconv(f,x)
% Inputs
%   f   filter
%   x   1-d signal
% Outputs
%   y   filtered result
%
% Description
%   Filtering by periodic convolution of x with f
%
% See Also
%   aconv, UpDyadHi, UpDyadLo, DownDyadHi, DownDyadLo
%
n = length(x);
p = length(f);
if p <= n,
  xpadded = [x((n+1-p):n) x];
else
  z = zeros(1,p);
  for i=1:p,
  imod = 1 + rem(p*n -p + i-1,n);
  z(i) = x(imod);
  end
  xpadded = [z x];
end
ypadded = filter(f,1,xpadded);
y = ypadded((p+1):(n+p));

%
% Copyright (c) 1993. David L. Donoho
%


%
% Part of WaveLab Version .700
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

loadgrayref.m

```
%%%%%%%%%%%%%%%
%loadgrayref loads the grayscale reference images
%%%%%%%%%%%%%%%
for k = 1:5;
```

```
for j = 1:8;
if k == 5 & j > 4;
break;
elseeval(['load grayref' num2str(k) num2str(j)]);
end;
end;
end;
```

loadgraytest.m

```
%%%%%%%%%%%%%%
%loadgrayref loads the grayscale reference images
%%%%%%%%%%%%%%
for k = 1:4;
for j = 1:13;
if exist(['grayref' num2str(k) num2str(j) '.mat']) == 0;
break;
elseeval(['load grayref' num2str(k) num2str(j)]);
end;
end;
end;
```

MakeONFilter.m

```
function f = MakeONFilter(Type,Par)
% MakeONFilter -- Generate Orthonormal QMF Filter for Wavelet Transform
% Usage
%   qmf = MakeONFilter(Type,Par)
% Inputs
%   Type   string, 'Haar', 'Beylkin', 'Coiflet', 'Daubechies',
%          'Symmlet', 'Vaidyanathan'
%   Par    integer, e.g. if Type = 'Coiflet', Par=3 specifies
%          a Coiflet-3 wavelet
% Outputs
%   qmf    quadrature mirror filter
%
% Description
%   The Haar filter (which could be considered a Daubechies-2) was the
%   first wavelet, though not called as such, and is discontinuous.
%
%   The Beylkin filter places roots for the frequency response function
%   close to the Nyquist frequency on the real axis.
%
%   The Coiflet filters are designed to give both the mother and father
%   wavelets 2*Par vanishing moments; here Par may be one of 1,2,3,4 or 5.
%
%   The Daubechies filters maximize the smoothness of the father wavelet
%   (or "scaling function") by maximizing the rate of decay of its Fourier
%   transform. They are indexed by their length, Par, which may be one of
%   4,6,8,10,12,14,16,18 or 20.
%
%   Symmlets are the "least asymmetric" compactly-supported wavelets with
```

```
%  maximum number of vanishing moments, here indexed by Par, which ranges
%  from 4 to 10.
%
%  The Vaidyanathan filter gives an exact reconstruction, but does not
%  satisfy any moment condition. The filter has been optimized for
%  speech coding.
%
% See Also
%  FWT_PO, IWT_PO, FWT2_PO, IWT2_PO, WPAnalysis
%
% References
%  The books by Daubechies and Wickerhauser.
%

if strcmp(Type,'Haar'),
f = [1 1] ./ sqrt(2);
end

if strcmp(Type,'Beylkin'),
f = [.099305765374.424215360813.699825214057...
.449718251149-.110927598348-.264497231446...
.026900308804.155538731877-.017520746267...
-.088543630623.019679866044.042916387274...
-.017460408696-.014365807969.010040411845...
.001484234782-.002736031626.000640485329];
end

if strcmp(Type,'Coiflet'),
if Par==1,
f = [.038580777748-.126969125396-.077161555496...
.607491641386.745687558934.226584265197];
end
if Par==2,
f = [.016387336463-.041464936782-.067372554722...
.386110066823.812723635450.417005184424...
-.076488599078-.059434418646.023680171947...
.005611434819-.001823208871-.000720549445];
end
if Par==3,
f = [-.003793512864.007782596426.023452696142...
-.065771911281-.061123390003.405176902410...
.793777222626.428483476378-.071799821619...
-.082301927106.034555027573.015880544864...
-.009007976137-.002574517688.001117518771...
.000466216960-.000070983303-.000034599773];
end
if Par==4,
f = [.000892313668-.001629492013-.007346166328...
.016068943964.026682300156-.081266699680...
-.056077313316.415308407030.782238930920...
.434386056491-.066627474263-.096220442034...
```

```
.039334427123.025082261845-.015211731527...
-.005658286686.003751436157.001266561929...
-.000589020757-.000259974552.000062339034...
.000031229876-.000003259680-.000001784985];
end
if Par==5,
f = [-.000212080863.000358589677.002178236305...
-.004159358782-.010131117538.023408156762...
.028168029062-.091920010549-.052043163216...
.421566206729.774289603740.437991626228...
-.062035963906-.105574208706.041289208741...
.032683574283-.019761779012-.009164231153...
.006764185419.002433373209-.001662863769...
-.000638131296.000302259520.000140541149...
-.000041340484-.000021315014.000003734597...
.000002063806-.000000167408-.000000095158];
end
end

if strcmp(Type,'Daubechies'),
if Par==4,
f = [.482962913145.836516303738...
.224143868042-.129409522551];
end
if Par==6,
f = [.332670552950.806891509311...
.459877502118-.135011020010...
-.085441273882.035226291882];
end
if Par==8,
f = [ .230377813309.714846570553...
.630880767930-.027983769417...
-.187034811719.030841381836...
.032883011667-.010597401785];
end
if Par==10,
f = [.160102397974.603829269797.724308528438...
.138428145901-.242294887066-.032244869585...
.077571493840-.006241490213-.012580751999...
.003335725285];
end
if Par==12,
f = [.111540743350.494623890398.751133908021...
.315250351709-.226264693965-.129766867567...
.097501605587.027522865530-.031582039317...
.000553842201.004777257511-.001077301085];
end
if Par==14,
f = [.077852054085.396539319482.729132090846...
.469782287405-.143906003929-.224036184994...
.071309219267.080612609151-.038029936935...
```

```
-.016574541631.012550998556.000429577973...
-.001801640704.000353713800];
end
if Par==16,
f = [.054415842243.312871590914.675630736297...
.585354683654-.015829105256-.284015542962...
.000472484574.128747426620-.017369301002...
-.044088253931.013981027917.008746094047...
-.004870352993-.000391740373.000675449406...
-.000117476784];
end
if Par==18,
f = [.038077947364.243834674613.604823123690...
.657288078051.133197385825-.293273783279...
-.096840783223.148540749338.030725681479...
-.067632829061.000250947115.022361662124...
-.004723204758-.004281503682.001847646883...
.000230385764-.000251963189.000039347320];
end
if Par==20,
f = [.026670057901.188176800078.527201188932...
.688459039454.281172343661-.249846424327...
-.195946274377.127369340336.093057364604...
-.071394147166-.029457536822.033212674059...
.003606553567-.010733175483.001395351747...
.001992405295-.000685856695-.000116466855...
.000093588670-.000013264203];
end
end

if strcmp(Type,'Symmlet'),
if Par==4,
f = [-.107148901418-.041910965125.703739068656...
1.136658243408.421234534204-.140317624179...
-.017824701442.045570345896];
end
if Par==5,
f = [.038654795955.041746864422-.055344186117...
.281990696854 1.023052966894.896581648380...
.023478923136-.247951362613-.029842499869...
.027632152958];
end
if Par==6,
f = [.021784700327.004936612372-.166863215412...
-.068323121587.694457972958 1.113892783926...
.477904371333-.102724969862-.029783751299...
.063250562660.002499922093-.011031867509];
end
if Par==7,
f = [.003792658534-.001481225915-.017870431651...
.043155452582.096014767936-.070078291222...
```

```
.024665659489.7581626019641.085782709814...
.408183939725-.198056706807-.152463871896...
.005671342686.014521394762];
end
if Par==8,
f = [.002672793393-.000428394300-.021145686528...
.005386388754.069490465911-.038493521263...
-.073462508761.5153986703741.099106630537...
.680745347190-.086653615406-.202648655286...
.010758611751.044823623042-.000766690896...
-.004783458512];
end
if Par==9,
f = [.001512487309-.000669141509-.014515578553...
.012528896242.087791251554-.025786445930...
-.270893783503.049882830959.873048407349...
1.015259790832.337658923602-.077172161097...
.000825140929.042744433602-.016303351226...
-.018769396836.000876502539.001981193736];
end
if Par==10,
f = [.001089170447.000135245020-.012220642630...
-.002072363923.064950924579.016418869426...
-.225558972234-.100240215031.667071338154...
1.088251530500.542813011213-.050256540092...
-.045240772218.070703567550.008152816799...
-.028786231926-.001137535314.006495728375...
.000080661204-.000649589896];
end
end

if strcmp(Type,'Vaidyanathan'),
f = [-.000062906118.000343631905-.000453956620...
-.000944897136.002843834547.000708137504...
-.008839103409.003153847056.019687215010...
-.014853448005-.035470398607.038742619293...
.055892523691-.077709750902-.083928884366...
.131971661417.135084227129-.194450471766...
-.263494802488.201612161775.635601059872...
.572797793211.250184129505.045799334111];
end

f = f ./ norm(f);

%
% Copyright (c) 1993-5. Jonathan Buckheit and David Donoho
%


%
% Part of WaveLab Version .700
```

```
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

MirrorFilt.m

```
function y = MirrorFilt(x)
% MirrorFilt -- Apply (-1)^t modulation
% Usage
%   h = MirrorFilt(l)
% Inputs
%   l  1-d signal
% Outputs
%   h  1-d signal with DC frequency content shifted
%       to Nyquist frequency
%
% Description
%   h(t) = (-1)^(t-1)  * x(t),  1 <= t <= length(x)
%
% See Also
%   DyadDownHi
%

y = -( (-1).^(1:length(x)) ).*x;

%
% Copyright (c) 1993. Iain M. Johnstone
%


%
% Part of WaveLab Version .700
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

planetest.m

```
clear all;
loadplaneref;%Load reference images
loadplanetest;%Load test images
Pcorrplane1 = zeros(5,3);%Initialize overall recognition performance output variable
for trials = 1:12;%Run for 12 trials
Pcorr = zeros(5,3);%Initialize intermediate output variable
m = zeros(10,10);%Initialize Euclidean distance matrix
b = [4 12 20];%Run for all filter lengths
biterr = [.05 .02 .01 .001 .0001];%Run for all bit error rates
```

```
%%%%%%%%%%%%%%%%%%%%%%%%
%Main program - Compute Euclidean Distance between test and reference
%image feature vectors using wavelet coefficients as features
%%%%%%%%%%%%%%%%%%%%%%%%
for u2 = 1:length(biterr);
thisbiterr = biterr(u2);
for u = 1:length(b);
count2 = 1;%Initialize test image index
y = zeros(1,10);%Store identified plane here
qmf = MakeONFilter('Daubechies',b(u));%Create Daubechies wavelet filter of proper length
for k = 1:3;
for j = 1:4;

%Load test image
if exist(['testplane' num2str(k) num2str(j)]) == 0;
break;
else t = eval(['testplane' num2str(k) num2str(j)]);
count = 1;
for p = 1:3;

  %Load reference image
  for q = 1:4;
    if p == 3 & q > 2 ;
      break;
    else r = eval(['refplane' num2str(p) num2str(q)]);

      r = FWT2_PO(r,1,qmf);%Compute wavelet transform of
            %reference image
      r = reshape(r(1:128,1:128),1,128^2);%Store coefficients
              %as feature vector
      if count == 1;%If this is the first pass
        t = FWT2_PO(t,1,qmf);%Compute wavelet
              %transform of test
              %image
        t = reshape(t(1:128,1:128),1,128^2);%Store coefficients
              %as feature vector
        t = sig2bin(t,8,'uniform','gray');%Simulate digital
              %transmission
        %Introduce bit errors using a random number generator
        bitchange = find(rand(1,length(t)) < thisbiterr);
        noise = zeros(size(t));
        noise(bitchange) = ones(size(bitchange));

        %Corrupt transmitted vector with random bit errors
        t = xor(t,noise);
        t = bin2sig(t,8,'uniform','gray');%Decode the signal
      end;
      m(count2,count) = (r-t)*(r-t)';%Compute Euclidean Distance
      count = count + 1;%Increment reference image index
    end;
  end;
```

```
end;
end;
[n,j] = min(m(count2,:));%Determine index of reference image with smallest distance measure
y(count2) = j;%Convert to ASCII character
count2 = count2+1;%Increment test image index
end;
end;
ideal = 1:10;%Correct answer
Pcorr(u2,u) = 1 - length(find(y - ideal))/length(ideal);%Compute observed probability of correct
recognition


end;
end;
Pcorrplane1 = Pcorrplane1 + Pcorr;%Sum over all trials
save Pcorrplane1 Pcorrplane1;%save output variable
end;
end;
```

quadlength.m

```
function [n,J] = quadlength(x)
% quadlength -- Find length and dyadic length of square matrix
% Usage
%   [n,J] = quadlength(x)
% Inputs
%   x  2-d image; size(n,n), n = 2^J (hopefully)
% Outputs
%   n  length(x)
%   J  least power of two greater than n
%
% Side Effects
%   A warning message is issue if n is not a power of 2,
%   or if x is not a square matrix.
%
s = size(x);
n = s(1);
if s(2) ~= s(1),
  disp('Warning in quadlength: nr != nc')
end
k = 1 ; J = 0; while k < n , k=2*k; J = 1+J ; end ;
if k ~= n ,
  disp('Warning in quadlength: n != 2^J')
end

%
% Copyright (c) 1993. David L. Donoho
%


%
% Part of WaveLab Version .700
```

```
% Built Friday, December 8, 1995 8:36:37 PM
% This is Copyrighted Material
% For Copying permissions see COPYING.m
% Comments? e-mail wavelab@playfair.stanford.edu
%
```

transtest.m

```
%%%%%%%%%%%%%%%%%%%
%Determine performance of multiresolution image recognition scheme
%using Fourier transform coefficients as elements of feature vectors
%%%%%%%%%%%%%%%%%%%
clear all;
loadgrayref;%load reference images
loadgraytranstest;%load test images
%Initialize global variables for simulation
global START_OK;
START_OK = 1;

Pgraytrans1= zeros(5,3);%Initialize overall performance output variable
for trials = 1:12;%run for 12 trials

Pcorr = zeros(5,3);%Initialize intermediate output variable - percentage
%of correctly identified images for single trial
m = zeros(46,36);%Initialize Euclidean distance measure matrix
b = [4 12 20];%Length of Daubechies filter
biterr = [.05 .02 .01 .001 .0001];%Probability of bit error

%%%%%%%%%%%%%%%%%%%
%Main program: run simulation for all filter lengths and bit errors;
%%%%%%%%%%%%%%%%%%%
for u2 = 1:length(biterr);%Run for all bit errors
thisbiterr = biterr(u2);%P(b) for this trial
for u = 1:length(b);%Run for all filter lengths
count2 = 1;%Test image index
y = zeros(1,46);%Store text character identified in this vector
qmf = MakeONFilter('Daubechies',b(u));%Build Daubechies wavelet filter of given length

for k = 1:4;%Load test image
for j = 1:13;
if exist(['transtestgray' num2str(k) num2str(j) '.mat']) == 0;
break;
else    eval(['load transtestgray' num2str(k) num2str(j)]);
t = eval(['transtestgray' num2str(k) num2str(j)]);
%Compare with all reference images
count = 1;%reference image index
for p = 1:5;
  for q = 1:8;
    if p == 5 & q > 4;
      break;
```

```
    else r = eval(['grayref' num2str(p) num2str(q)]);
     r = FWT2_PO(r,1,qmf);%Compute wavelet transform of reference
            %image
     r = fftid(r);%Compute feature vector by taking
            %2 - dimensional Fourier transform
            %of each subband of wavelet transform
     if count == 1;%If this is the first reference image

       t = FWT2_PO(t,1,qmf);%Compute wavelet transform
              %of test image
       t = fftid(t);%Compute feature vector by taking
            %2 - dimensional Fourier transform
            %of each subband of wavelet transform

       t = quantize(t,8,'scale');%Quantize uniformly using
              %8 bits per coefficient

       t = bin_enc(t,8);%Encode using natural binary code
       t = bin2gray(t);%Convert to gray coding

     %Introduce random errors using random number generator with
     %probability = thisbiterr

       bitchange = find(rand(size(t)) < thisbiterr);
       noise = zeros(size(t));
       bitchange = find(rand(size(t)) < thisbiterr);
       noise(bitchange) = ones(size(bitchange));
     %Corrupt transmitted signal t with random bit errors found above
       t = xor(t,noise);
       t = gray2bin(t);%convert to natural binary code
       t = bin_dec(t);%decode signal
     end;
     m(count2,count) = (r-t)*(r-t)';%Compute distance between reference
              %and test vectors

     count = count + 1;
    end;
  end;
end;
end;
[n,j] = min(m(count2,:));%Determine index of reference image with minimum
      %Euclidean distance

y(count2) = num2let(j);%Convert to ASCII character

count2 = count2+1;%Increment test vector index

end;
end;
%Compare results obtained this trial with correct answer
```

```
ideal = ['2' '0' '9' '5' '6' 'Q' 'U' 'I' 'C' 'K' 'F' 'O' 'X' 'E' 'S' 'J' 'U' 'M' 'P' 'E' 'D' 'O' 'V' 'E'
'R' 'T' 'H' 'E' '1' '3' '4' '8' '7' 'L' 'A' 'Z' 'Y' 'B' 'R' 'O' 'W' 'N' 'D' 'O' 'G' 'S'];
Pcorr(u2,u) = 1 - length(find(y - ideal))/length(ideal);%Determine error performance


end;
end;
Pgraytrans1 = Pgraytrans1 + Pcorr;%Determine overall performance as sum of all trials
save Pgraytrans1 Pgraytrans1;%Store overall performance
end;
```

wavetest.m

```
%%%%%%%%%%%%%%%%%
%Test the performance of multiresolution image recognition over noisy channel
%using wavelet coefficients as feature vectors
%%%%%%%%%%%%%%%%%
clear all;
global START_OK;%Set global variable for transmission channel simulation
START_OK = 1;

loadgrayref;%load reference images
loadgraytest;%load test images
Pcorr1a = zeros(5,3);%initialize output variable - Probability of correct recognition overall


for trials = 1:12;%Perform test 12 times
Pcorr = zeros(5,3);%Intermediate output variable - Probability of correct recognition this trial
m = zeros(46,36);%Euclidian Distance measure matrix
b = [4 12 20];%Length of Daubechies filters to test
biterr = [ .05 .02 .01 .001 .0001];%Probability of bit errors to test

%%%%%
%Main part of program: determine recognition performance for each bit error length and
%each filter length at this resolution level (R1)

for u2 = 1:length(biterr);%Run for all bit errors
thisbiterr = biterr(u2);%P(b) this trial
for u = 1:length(b);%Run for all filter lengths
count2 = 1;%Index of test character
y = zeros(1,46);%Store text character identified in this vector
qmf = MakeONFilter('Daubechies',b(u));%Create Daubechies filter of correct length for this trial

for k = 1:5;%load next test character
for j = 1:13;
if exist(['graytest' num2str(k) num2str(j)]) == 0;
break;
else   t = eval(['graytest' num2str(k) num2str(j)]);

count = 1; %Compute Euclidean distance to all reference characters
for p = 1:5;
  for q = 1:8;
    if p == 5 & q > 4 ;
```

```
      break;
    else r = eval(['grayref' num2str(p) num2str(q)]);

      r = FWT2_PO(r,1,qmf);%Compute wavelet decompostion using
            %wavelet filter qmf

      r = reshape(r(1:128,1:128),1,128^2);%Reshape all
              %coefficients into
              %vector
     if count == 1;%If this is the first reference character,
         %compute the wavelet transform of the
         %test image and
         %simulate tranmitting it over a
         %noisy channel

       t = FWT2_PO(t,1,qmf);%Compute wavelet
            %decomposition using qmf
       t = reshape(t(1:128,1:128),1,128^2);%Put into vector
       t = quantize(t,8,'scale');%Quantize uniformly using
            %8 bits per coefficient
       t = bin_enc(t,8);%Use natural binary encoding
       t = bin2gray(t);%convert to gray coding

       %Introduce bit errors randomly with probability
       %equal to thisbiterr using random number
       %generator
       bitchange = find(rand(1,length(t)) < thisbiterr);
       noise = zeros(size(t));
       noise(bitchange) = ones(size(bitchange));
       %Corrupt t with random errors from abover
       t = xor(t,noise);
       %Recover corrupted signal
       t = gray2bin(t);%convert to natural binary
       t = bin_dec(t);%Decode signal
     end;
     m(count2,count) = (r-t)*(r-t)';%Compute Euclidean distance
     count = count + 1;%Increment reference image index
    end;
  end;
 end;
 end;
 [n,j] = min(m(count2,:));%Determine index of smallest Euclidean distance

 y(count2) = num2let(j);%Convert to ASCII letter
 count2 = count2+1;%Increment test image index
 end;
 end;

 %%Compare results with correct answer
 ideal = ['2' '0' '9' '5' '6' 'Q' 'U' 'I' 'C' 'K' 'F' 'O' 'X' 'E' 'S' 'J' 'U' 'M' 'P' 'E' 'D' 'O' 'V' 'E'
'R' 'T' 'H' 'E' '1' '3' '4' '8' '7' 'L' 'A' 'Z' 'Y' 'B' 'R' 'O' 'W' 'N' 'D' 'O' 'G' 'S'];
```

```
Pcorr(u2,u) = 1 - length(find(y - ideal))/length(ideal);%Determine error performance

end;
end;
Pcorr1a = Pcorr1a + Pcorr%Determine overall performance as sum of individual trials
trials
save Pcorr1a Pcorr1a;%Save output
end;
end;
```

LIST OF REFERENCES

1. Gonzalez, Rafael C., and Woods, Richard E., *Digital Image Processing,* Addison Wesley Publishing Company, Inc., Reading, MA, 1992.

2. Pratt, William K., *Digital Image Processing,* John Wiley & Sons, New York, NY, 1978.

3. Castleman, Kenneth R., *Digital Image Processing,* Prentice Hall, Inc., Englewood Cliffs, NJ, 1996.

4. Cohen, Albert, and Kovacevic, Jelena, "Wavelets: The Mathematical Background." *Proceedings of the IEEE.* Vol. 84, No. 4, April 1996.

5. Stollnitz, Eric J. Derose, Tony D., and Salesin, David H., "Wavelets for Computer Graphics: A Primer." *IEEE Computer Graphics and Applications,* May 1995.

6. Burrus, C. S., and Gopinath, R. A., *Introduction to Wavelets and Wavelet Transforms.* Rice University, Houston, TX, 1993.

7. Kaiser, Gerald, *A Friendly Guide to Wavelets.* Birkhauser, Boston, 1994.

8. Proakis, John G., and Manolakis, Dimitris G., *Digital Signal Processing,* Macmillan Publishing Company. New York, NY, 1992.

9. Carvalho, Ronald, *Mulitresolution Image Compression Using Subband Coding and Wavelet Decompositon,* Engineer's Thesis, Naval Postgraduate School, Monterey, December, 1994.

10. Leon, Steven J., *Linear Algebra With Applications,* Macmillan College Publishing Company, New York, NY, 1994.

11. Akansu, Ali N., and Haddad, Richard A., *Multiresolution Signal Decomposition,* Academic Press, Inc., Boston, MA, 1992.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944, Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School 411 Dyer Rd. Monterey, CA 93943-5101	2
3.	Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
4.	Prof. Murali Tummala, Code EC/Tu Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	4
5.	Prof. Ralph Hippenstiel, Code EC/Hi Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5121	1
6.	LCDR Gregory H. Skinner Officer-In-Charge Patrol Wings Atlantic Detachment Advanced Maritime Projects Office Jacksonville, FL 32212-0051	1
7.	LT William M. Peyton, Jr. 1848 Cherokee Dr. #6. Salinas, CA 93906	2